

# Feature-driven Emergence of Model Graphs for Object Recognition and Categorization

Günter Westphal<sup>a</sup>, Christoph von der Malsburg<sup>a,b,c</sup>, Rolf P. Würtz<sup>a,\*</sup>

<sup>a</sup>Institut für Neuroinformatik  
Ruhr-Universität Bochum  
D-44780 Bochum, Germany

<sup>b</sup>Laboratoy for Computational and Biological Vision  
University of Southern California  
Los Angeles, 90089-2520, USA

<sup>c</sup>Frankfurt Institute for Advanced Studies  
Johann Wolfgang Goethe-Universität Frankfurt am Main  
D-60438 Frankfurt am Main, Germany

9th February 2006

---

\*Corresponding author: [rolf.wuertz@neuroinformatik.rub.de](mailto:rolf.wuertz@neuroinformatik.rub.de)

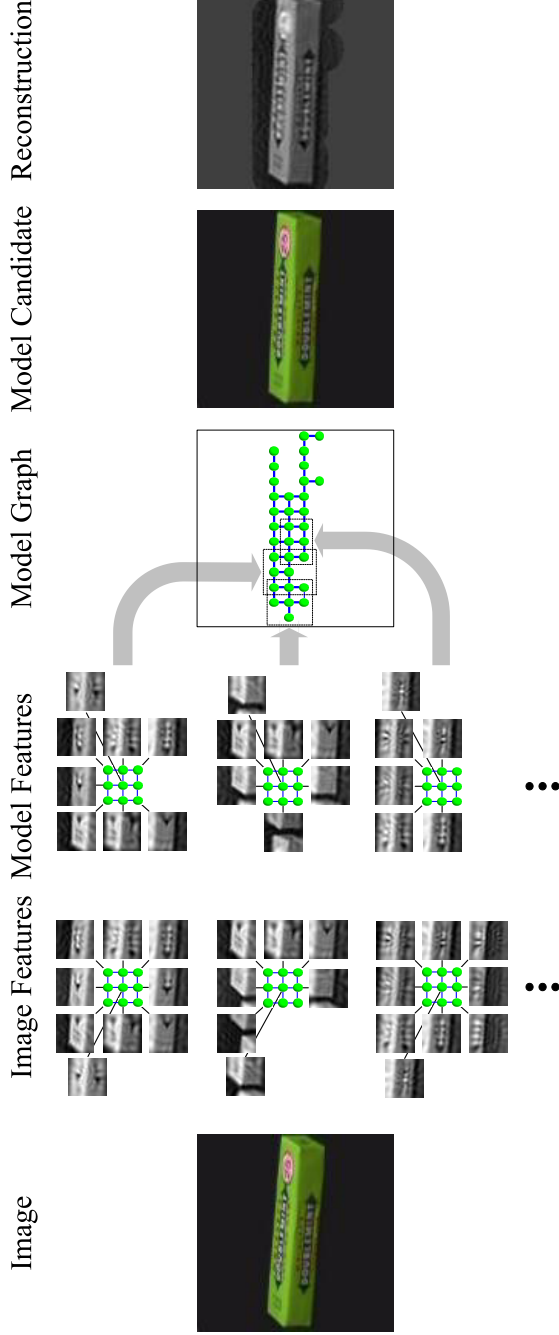
# 1 Introduction

An important requirement for the expression of cognitive structures is the ability to form mental objects by rapidly binding together constituent parts [2, 3]. In this sense, one may conceive the brain’s data structure to have the form of graphs whose nodes are labeled with elementary features.

This data format has been used for visual object recognition [26, 4, 5, 16], and in the Dynamic Link Matching approach [32, 33, 34, 9, 40]. In all these approaches the data structure of stored objects has the form of graphs whose nodes are labeled with elementary features. These are called *model graphs* and provide a view-tuned representation [19, 21] of the object contained in the presented image. They provide a versatile data format with the capability to render the structure of any object. Because of the multitude of possible object variations like changes in identity, pose, or illumination, the graphs are required to be dynamic with respect both to shape and attributed features.

Upon presentation of an image a so-called model graph should rapidly emerge by binding together memorized subgraphs derived from earlier learning examples driven by the image features. Emergence of model graphs is a laborious task which, in computer vision, has most often been disregarded in favor of employing model graphs tailored to specific object categories like, faces in frontal pose [9, 42, 40]. Recognition or categorization of arbitrary objects, however, demands dynamic graphs, i.e., more emphasis must be laid on the question of how model graphs are created from raw image data.

Relatively little work has been done on the dynamic creation of model graphs. The object recognition system proposed in [35] is based on Dynamic Link Matching supplied with object memory. While learning novel objects a so-called fusion graph is created through iteratively matching image graphs with the fusion graph and grafting non-matched parts of image graphs into the fusion graph. When an object is to be recognized, one or more image graphs are compared against model memory via graph matching, implemented by dynamic links. The matching parts of the fusion graph thus constitute the model graph for the object contained in the input image. The system has proven to perform well for a small number of object views. During both learning and recognition the objects are required to be placed in front of a plain background.



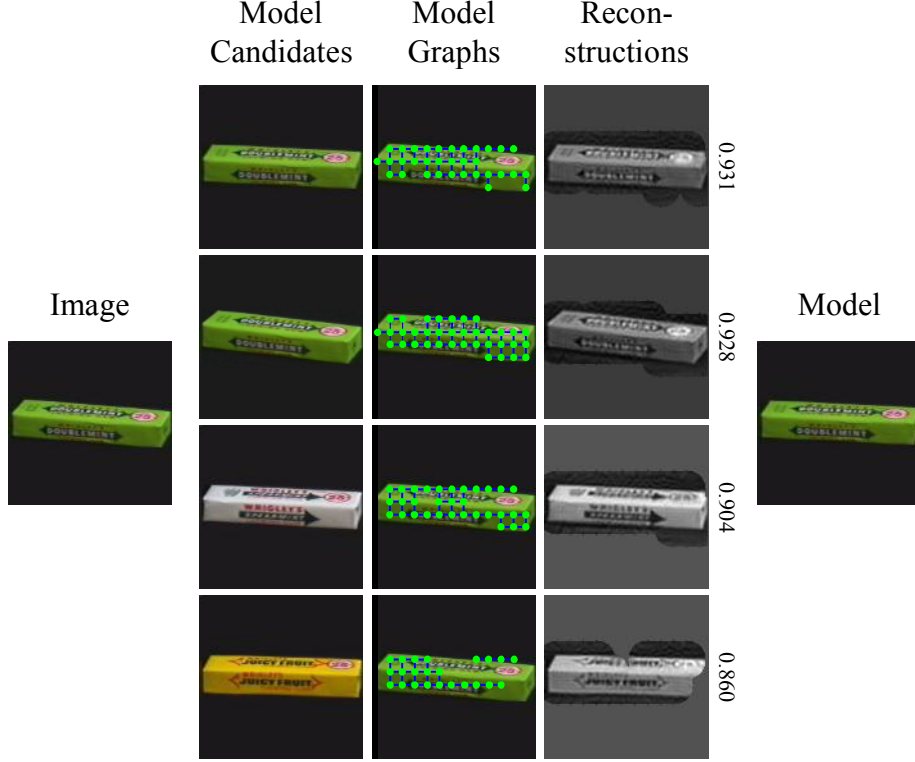
**Figure 1: Feature-driven Emergence of Model Graphs** — Upon presentation of an image (first column) a model graph (fourth column) should rapidly emerge by binding together (arrows) memorized parquet graphs derived from earlier learning examples (third column) that match with the image features (second column). Column six shows the reconstruction from the model graph. The graph dynamics itself proceeds in three steps. In the first step position-invariant feature detectors, are learned from training images. Through combination of these classifiers into a single-layer perceptron, a weighted majority voting scheme is implemented. It allows for preselection of salient learning examples, so-called model candidates (fifth column), and likewise for preselection of salient categories the object in the presented image belongs to. Each model candidate is verified in a third step using a variant of elastic graph matching. To further differentiate between model candidates with similar features in the same spatial arrangement for the model features is asserted. Reconstruction and the model candidate contain the same object in the same pose, which is slightly different from the one in the input image.

A different approach is the creation of model graphs with minimal user-assistance [13]. In that method, a growing neural gas [7] is used to determine shape and topology of a model graph. Binarized difference images derived from two consecutive images of the same moving object are used as an input to a growing neural gas, whose nodes are attracted to superthreshold frame differences. Upon an user-initiated event, Gabor jets are extracted at the node positions and the produced model graph is stored in a model database. During recognition, model graphs are matched in succession with the input image. The compositional aspect is thus prominent while learning novel objects but is absent during recognition. A rudimentary version of model graph dynamics is also present in [42], where model graphs are adapted to segmentation masks in order to ignore background influences.

Recognition methods relying on graph matching are *correspondence-based* in the sense that image point correspondences are estimated before recognition is attempted. This estimation is usually only possible on the basis of the spatial arrangement of elementary features. There is also a class of recognition algorithms, which are purely *feature-based* and completely disregard feature arrangement. A prominent example is SEEMORE [15]. There it is shown that a simple neural network can distinguish objects in a purely feature-based way if enough feature types are employed. As a model for recognition and categorization in the brain, feature-based methods can be implemented as feedforward networks, which would account for the amazing speed with which these processes can be carried out, relative to the slow processing speed of the underlying neurons [28, 29]. These methods, however encounter problems in the case of multiple objects and highly structured backgrounds. From the point of view of pattern recognition, feature-based methods are *discriminative* while graph matching is *generative* [30].

It is reasonable to assume that feedforward processing is applied as far as it goes by excluding as many objects as possible and that only ambiguous cases are subjected to correspondence-based processing, which is more time-consuming.

In this chapter we propose a form of graph dynamics, which proceeds in three steps. In the first step *position-invariant feature detectors*, which decide whether a feature is present in an image, are learned from training images. For processing arbitrary objects, features are small localized grid graphs, so-called *parquet graphs*, whose nodes are attributed with Gabor amplitudes. Through combination of these classifiers into a single layer perceptron that conforms to Linsker's infomax principle, the so-called *pre-selection network*, a weighted majority voting scheme [10] is implemented. It allows for preselection of salient learning examples, so-called *model candidates*, and likewise for preselection of salient categories the object in the presented image supposedly belongs to. Each model candidate is verified



**Figure 2: Selection of the Model** — Given the input image in the first column, the preselection network selects four model candidates (second column). As has been illustrated in fig. 1, a model graph is dynamically constructed for each model candidate by assembling matching model features into larger graphs according to their spatial arrangement (third column). The fourth column shows the reconstruction from each model graph. Each model candidate is verified using a rudimentary version of elastic graph matching. Model graphs are optimally placed on the object contained in the input image in terms of maximizing the measure of similarity (third column). The attained similarities between the model candidates, represented by their model graphs, and the input image are annotated to the reconstructions. The model candidate that yields the best similarity to the input image is chosen as the recognized model (fifth column).

in a third step using a rudimentary version of elastic graph matching. To further differentiate between model candidates with similar features it is asserted that the features be in similar spatial arrangement for the model to be selected. In this way model graphs are constructed dynamically by assembling model features into larger graphs according to their spatial arrangement (Fig. 1). Finally, the resulting model graphs are matched with a rudimentary version of elastic graph matching, and the model candidate that yields the best similarity to the input image is chosen as the recognized model (Fig. 2).

The description of the method is accompanied by a *case study*, which exemplifies the various steps on an example, in which only two images of two objects are learned and distinguished.

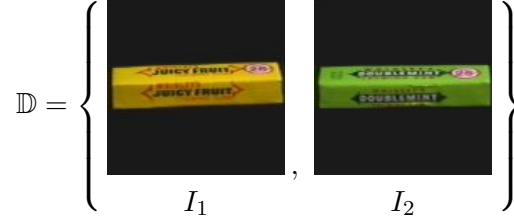
## 2 Learning Set, Partitionings, and Categories

There are many different classifications that can be made on image data. For object recognition, all instances of the same object under different pose and/or illumination are to be put into the same class. An alternative learning problem may be the classification of illumination or pose regardless of object identity. A hallmark of human visual cognition is the classification into *categories*: we group together images of cats, dogs, insects, and reptiles into the category 'animal' and are able to differentiate animals from non-animals with impressive speed [28].

Following [18] we use the term *recognition* for a decision about an object's unique identity. Recognition thus requires subjects to discriminate between similar objects and involves generalization across some shape changes as well as physical translation, rotation and so forth. The term *categorization* refers to a decision about an object's kind. Categorization thus requires generalization across members of a class of objects with different shapes. Especially, the system has to generalize over object identity.

We start by considering some finite set of images  $\mathbb{I}$  and a subset  $\mathbb{D}$ , which we call the *learning set*. In our case study the learning set comprises two images of different chewing gum packages in approximately the same pose (Fig. 3).

In order to accommodate the various learning tasks that can be imposed on a single image set we consider that there exist  $K$  *partitionings*  $\Pi^k$  of the



**Figure 3: Case Study: Learning Set** — *The learning set comprises two images of different chewing gum packages in approximately the same pose. The images are taken from the COIL-100 database [17]. In the following these images are referred to as  $I_1$  and  $I_2$ .*

learning set (Eq. (1)). A partitioning  $\Pi^k$  consists of  $C^k$  pairwise disjoint partitions  $\mathbb{C}_c^k$ .

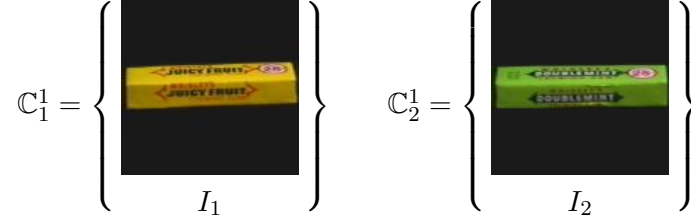
$$\begin{aligned} \Pi^k &= \{ \mathbb{C}_c^k \subseteq \mathbb{D} \mid 1 \leq c \leq C^k \} \\ \text{with } \forall c \neq c' : \mathbb{C}_c^k \cap \mathbb{C}_{c'}^k &= \emptyset \quad \text{and} \quad \bigcup_{c=1}^{C^k} \mathbb{C}_c^k = \mathbb{D} \end{aligned} \quad (1)$$

The objects in the images of a particular partition are conceived to share a common semantic property, for instance, being images of animals, or having the same illumination direction. Accordingly, partitions in the following are termed *categories*. *Category labels*  $c$  range between 1 and  $C^k$ ; their range implicitly depends on the number of categories in the underlying partitioning  $\Pi^k$ . For simultaneous recognition of the object’s identity and the object’s pose the learning set is subdivided into single-element categories while for object categorization purposes the learning set is usually organized in a hierarchy of categories. Fig. 4 shows the single partitioning of the learning set in our case study.

A hierarchical categorization task can be exemplified with the ETH-80 image database [11]. That database comprises images of apples, pears, tomatoes, dogs, horses, cows, cups, and cars in varying poses and identities and has been used for the categorization experiments in 7.2. For those experiments we created  $K = 3$  partitionings of the learning set as shown in fig. 5.

### 3 Parquet Graphs

The feature-based part of the technique described in this paper can work with any convenient feature type. A successful application employing color



**Figure 4: Case Study: Partitioning of the Learning Set** — *In our case study there exists only  $K = 1$  partitioning  $\Pi^1$  of the learning set (Fig. 3). The partitioning consists of  $C^1 = 2$  single-element categories  $\mathbb{C}_1^1 = \{I_1\}$  and  $\mathbb{C}_2^1 = \{I_2\}$ .*

and multiresolution image information is presented in [38]. For the current combination of feature- and correspondence-based methods we chose small regular graphs labeled with Gabor features. We call them *parquet graphs* inspired by the look of ready-to-lay parquet tiles. These can work as simple feature detectors for preselection and be aggregated to larger graph entities for correspondence-based processing.

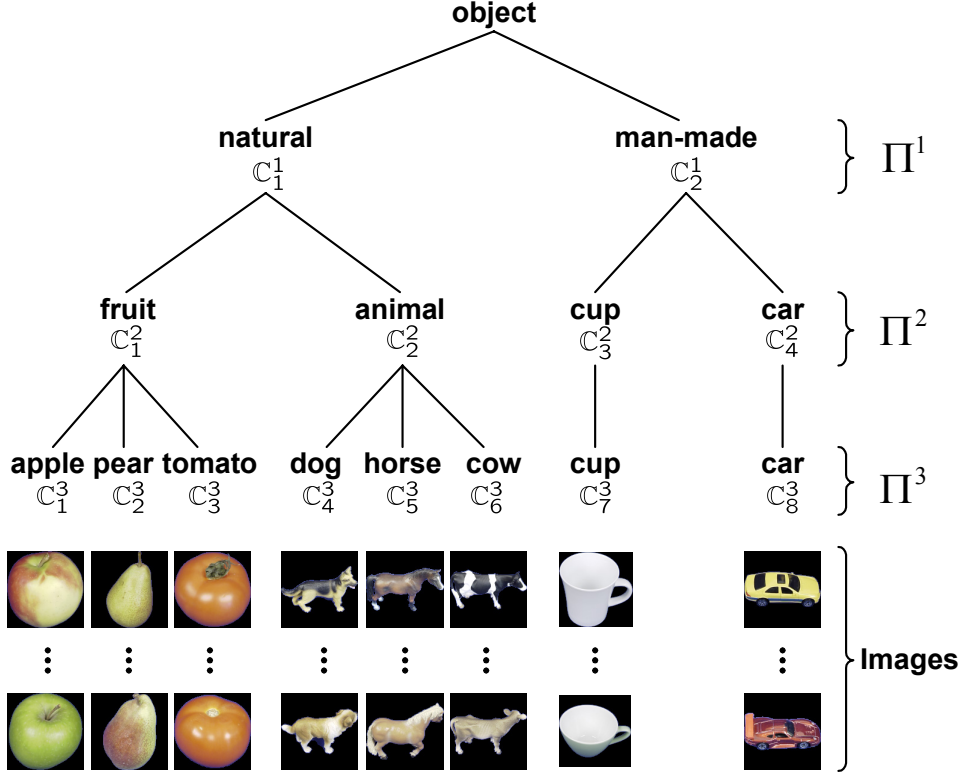
Throughout this paper, parquet graphs are constituted out of  $V = 9$  nodes. In the following, a parquet graph  $f$  is described with a finite set of node attributes: Each node  $v$  is labeled with a triple  $(\underline{x}_v, \mathcal{J}_v, b_v)$  where  $\mathcal{J}_v$  is a Gabor jet derived from an image at an absolute node position  $\underline{x}_v$ . Computation and parameters of the Gabor features is the same as in [9, 40]. In order to make use of segmentation information it is convenient to mark certain nodes as *invalid* and exclude them from further calculation in that way. For this purpose the node attributes comprise the validity flag  $b_v$  that can take the values 0 and 1, meaning ‘invalid’ and ‘valid’. The horizontal and vertical node distances  $\Delta x$  and  $\Delta y$  are set to 10 pixels in this work.

$$f = \{(\underline{x}_v, \mathcal{J}_v, b_v) | 1 \leq v \leq V\} \quad (2)$$

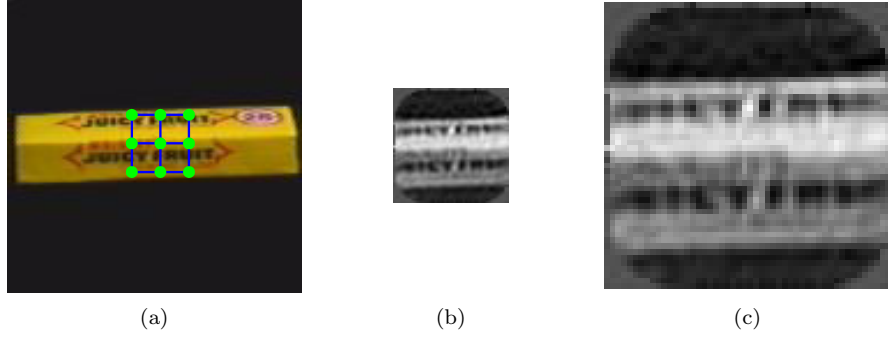
Fig. 6 shows an example of a parquet graph that has been placed on the object in learning image  $I_1$ . Where appropriate, instances of parquet graphs are, more generally, called features or feature instances.

For selection of salient categories and model candidates, the feature-based part of the proposed system, a parquet graph describes a patch of texture derived from an image regardless of its position in the image plane. Particularly, this means that the node positions are irrelevant for the decision whether two images contain a similar patch of texture. Later, for verification of the selected model candidates, i.e., learning images that may serve





**Figure 5: Hierarchical Organization of Categories** — A hierarchy of categories on the ETH-80 image database [11], which contains images of apples, pears, tomatoes, dogs, horses, cows, cups, and cars in varying poses and identities. We created  $K = 3$  partitionings  $\Pi^1, \Pi^2$ , and  $\Pi^3$ . Partitioning  $\Pi^1$  comprises  $C^1 = 2$  categories of natural ( $C_1^1$ ) and man-made objects ( $C_2^1$ ). Partitioning  $\Pi^2$  comprises  $C^2 = 4$  categories of fruits ( $C_1^2$ ), animals ( $C_2^2$ ), cups ( $C_3^2$ ), and cars ( $C_4^2$ ). Finally, partitioning  $\Pi^3$  comprises  $C^3 = 8$  categories of apples ( $C_1^3$ ), pears ( $C_2^3$ ), tomatoes ( $C_3^3$ ), dogs ( $C_4^3$ ), horses ( $C_5^3$ ), cows ( $C_6^3$ ), cups ( $C_7^3$ ), and cars ( $C_8^3$ ).



**Figure 6: Example of a Parquet Graph** — *Figure (a) shows a parquet graph that has been placed on the object in learning image  $I_1$ . Each node of a parquet graph is attributed with Gabor amplitudes derived from an image at the node’s position. Figure (b) shows the reconstruction from the parquet graph. Figure (c) is an enlarged version of figure (b). The reconstruction is computed with the algorithm from [20].*

---

as models for the input image, larger graphs are constructed dynamically by assembling parquet graphs derived from earlier learning images according to their spatial arrangement. Thus, within the correspondence-based part, the node positions will become important.

### 3.1 Similarity Function

The measure of similarity between two parquet graphs  $f$  and  $f'$  is defined as the normalized sum of the similarities between valid Gabor jets [42, 24] attached to nodes with the same index that stem from the given parquet graphs (Eq. (4)). Throughout this paper, the similarity between two Gabor jets is given by the normalized scalar product between the absolute values of the complex components of the two jets (Eq. (3)). Let  $a_n$  denote the absolute value of  $n$ -th filter response.

$$s_{abs}(\mathcal{J}, \mathcal{J}') = \frac{\sum_n a_n a'_n}{\sqrt{\sum_n a_n^2 \cdot \sum_n a_n'^2}} \quad (3)$$

By definition, the factors  $(b_v b'_v)$  are 1 if the respective jets  $\mathcal{J}_v$  and  $\mathcal{J}'_v$  have both been marked as valid, and 0 otherwise. Thus, these factors assert that only similarities between jets that have both been marked as valid are

taken into account. If all products become 0, the similarity between the two parquet graphs yields 0.

$$s_{graph}(f, f') = \begin{cases} \left( \sum_{v=1}^V b_v b'_v \right)^{-1} \cdot \sum_{v=1}^V (b_v b'_v) \cdot s_{abs}(\mathcal{J}_v, \mathcal{J}'_v) & \text{if } \sum_{v=1}^V b_v b'_v > 0 \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

From the viewpoint of the correspondence problem, two parquet graphs in different images establish a local *field* of contiguous point-to-point correspondences. The similarity measure assesses how well points in two images specified by the given parquet graphs actually correspond to each other. It is well worth noting that parquet graphs provide a means to protect from accidentally establishing point-to-point correspondences in that contiguous, topographically smooth fields of good correspondences are favored over good but topographically isolated ones.

### 3.2 Local Feature Detectors

For the assessment whether two parquet graphs  $f$  and  $f'$  convey similar patches of texture with respect to a given sensitivity profile we introduce local feature detectors that return 1 if the similarity between the given parquet graphs is greater or equal than a given similarity threshold  $0 < \vartheta \leq 1$ , and 0 otherwise (Eq. (5)). We say that two parquet graphs *match* with respect to a given similarity threshold if the local feature detector returns 1.

$$\varepsilon(f, f', \vartheta) = \begin{cases} 1 & \text{if } s_{graph}(f, f') \geq \vartheta \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

Matching features are one argument for point-to-point *correspondences*, which needs to be backed up by the spatial arrangement of several matching features.

## 4 Learning a Visual Dictionary

Our goal is to formulate a graph dynamics that, upon image presentation, lets a model graph rapidly emerge by binding together memorized subgraphs derived from earlier learning examples. To that end we need to compute a stock of parquet graphs from learning examples in advance. These play the role of a visual dictionary. Parquet graphs derived from an input image during classification are looked up in the dictionary to find out which image and model features match. Each coincidence of a matching feature in the

image and model domain may then be considered as a piece of evidence that the input image belongs to the same categories as the learning image which contains the model feature.

#### 4.1 Feature Calculators

In eq. (6) we define  $R$  functions  $f^r$  capable of extracting a set of features out of an image. Let  $\mathbb{F}$  denote the set of all possible features. In the following these functions will be called *feature calculators*. The index  $r$  implicitly specifies the parameterization of the parquet graphs returned from the respective feature calculator  $f^r$ , like the similarity threshold  $\vartheta^r$ , which is employed in the local feature detectors. Generally, feature calculators are not restricted to parquet graphs; other types of features have been used in [38, 23, 36, 1].

$$f^r : \mathbb{I} \rightarrow \wp(\mathbb{F}) \text{ with } r \in \{1, \dots, R\} \quad (6)$$

For extraction of parquet graphs, the inter-node distances  $\Delta x$  and  $\Delta y$  are also used to specify a grid in the image plane. At each grid position allowing for placement of a whole parquet graph a parquet graph is extracted. Scanning of the image starts in the upper left corner from left to right to the lower right corner. If the image is known to be figure-ground segmented, parquet graphs with the majority of nodes residing in the background will be disregarded, the others have background points marked as invalid.

In the case study, we employ only  $R = 1$  feature calculator  $f^1$ . The feature calculator returns a set of parquet graphs with ten pixels distance between two neighbored nodes in horizontal and in vertical direction, respectively. Fig. 7 shows the result of consecutively applying this feature calculator to both learning examples.

#### 4.2 Feature Vectors

Looking at the number of parquet graphs that have been extracted from just two images (Fig. 7) it is clear that for learning sets with thousands or even ten thousands of images the total number of features would grow into astronomical dimensions. Consequently, we have to limit the total number of features to a tractable number. For this task we employ a simple variant of vector quantization [8], a lossy data compression technique. The applied vector quantization method is shown as pseudo code in fig. 8. In this fashion, each of the  $R$  feature calculators is used to compute a feature vector  $\underline{f}^r$  with  $r \in \{1, \dots, R\}$ . In the following  $T^r$  denotes the number of features in feature vector  $\underline{f}^r$ . All  $R$  feature vectors constitute the visual dictionary.

$$\begin{aligned}
f^1 \left( \begin{array}{c} \text{Image of a yellow Juicy Fruit gum pack} \\ I_1 \end{array} \right) &= \left\{ \begin{array}{l} \begin{array}{c} \text{Thumbnail (1,1)} \\ \text{Thumbnail (1,2)} \\ \text{Thumbnail (1,3)} \\ \text{Thumbnail (1,4)} \\ \text{Thumbnail (1,5)} \\ \text{Thumbnail (1,6)} \end{array} \\ \begin{array}{c} \text{Thumbnail (1,7)} \\ \text{Thumbnail (1,8)} \\ \text{Thumbnail (1,9)} \\ \text{Thumbnail (1,10)} \\ \text{Thumbnail (1,11)} \\ \text{Thumbnail (1,12)} \end{array} \\ \begin{array}{c} \text{Thumbnail (1,13)} \\ \text{Thumbnail (1,14)} \\ \text{Thumbnail (1,15)} \\ \text{Thumbnail (1,16)} \\ \text{Thumbnail (1,17)} \\ \text{Thumbnail (1,18)} \end{array} \\ \begin{array}{c} \text{Thumbnail (1,19)} \\ \text{Thumbnail (1,20)} \\ \text{Thumbnail (1,21)} \\ \text{Thumbnail (1,22)} \\ \text{Thumbnail (1,23)} \\ \text{Thumbnail (1,24)} \end{array} \\ \begin{array}{c} \text{Thumbnail (1,25)} \\ \text{Thumbnail (1,26)} \\ \text{Thumbnail (1,27)} \\ \text{Thumbnail (1,28)} \\ \text{Thumbnail (1,29)} \\ \text{Thumbnail (1,30)} \end{array} \\ \begin{array}{c} \text{Thumbnail (1,31)} \end{array} \end{array} \right\} \\
\\
f^1 \left( \begin{array}{c} \text{Image of a green Doublemint gum pack} \\ I_2 \end{array} \right) &= \left\{ \begin{array}{l} \begin{array}{c} \text{Thumbnail (2,1)} \\ \text{Thumbnail (2,2)} \\ \text{Thumbnail (2,3)} \\ \text{Thumbnail (2,4)} \\ \text{Thumbnail (2,5)} \\ \text{Thumbnail (2,6)} \end{array} \\ \begin{array}{c} \text{Thumbnail (2,7)} \\ \text{Thumbnail (2,8)} \\ \text{Thumbnail (2,9)} \\ \text{Thumbnail (2,10)} \\ \text{Thumbnail (2,11)} \\ \text{Thumbnail (2,12)} \end{array} \\ \begin{array}{c} \text{Thumbnail (2,13)} \\ \text{Thumbnail (2,14)} \\ \text{Thumbnail (2,15)} \\ \text{Thumbnail (2,16)} \\ \text{Thumbnail (2,17)} \\ \text{Thumbnail (2,18)} \end{array} \\ \begin{array}{c} \text{Thumbnail (2,19)} \\ \text{Thumbnail (2,20)} \\ \text{Thumbnail (2,21)} \\ \text{Thumbnail (2,22)} \\ \text{Thumbnail (2,23)} \\ \text{Thumbnail (2,24)} \end{array} \\ \begin{array}{c} \text{Thumbnail (2,25)} \end{array} \end{array} \right\}
\end{aligned}$$

**Figure 7: Case Study: Application of the Feature Calculator to the Learning Images** — *The thumbnail images in the returned sets on the right hand side are reconstructions from the extracted parquet graphs. Each reconstruction is uniquely labeled with a tuple. The first component addresses the learning image the parquet graph stems from while the second component is a sequential number.*

**Algorithm 1:** *vectorQuantization*

**Parameter:** Learning Set:  $\mathbb{D}$   
**Parameter:** Feature Calculator:  $f^r : \mathbb{I} \rightarrow \wp(\mathbb{F})$   
**Parameter:** Similarity Threshold:  $\vartheta^r$ ;  $0 < \vartheta^r \leq 1$   
**Result** : Feature Vector of Length  $T^r$ :  $\underline{f}^r$

```

1  $\mathbb{F}^r \leftarrow \emptyset$ 
2  $T^r \leftarrow 0$ 

3 forall  $I \in \mathbb{D}$  do
4   forall  $f \in f^r(I)$  do
5     if  $\forall f' \in \mathbb{F}^r : \varepsilon(f, f', \vartheta^r) = 0$  then
6        $\mathbb{F}^r \leftarrow \mathbb{F}^r \cup \{f\}$ 
7        $T^r \leftarrow T^r + 1$ 
8     end
9   end
10 end

11  $\underline{f}^r =: (f_t^r)_{1 \leq t \leq T^r} \leftarrow (0)_{1 \leq t \leq T^r}$ 
12  $t \leftarrow 0$ 
13 forall  $f \in \mathbb{F}^r$  do
14    $f_t^r \leftarrow f$ 
15    $t \leftarrow t + 1$ 
16 end

17 return  $\underline{f}^r$ 

```

**Figure 8: Vector Quantization Method** — The algorithm computes a codebook of so-called codebook vectors. In our case parquet graphs become employed as codebook vectors while the codebook is a set of parquet graphs. The size of the feature set depends considerably on the value of the similarity threshold  $\vartheta^r$ . For lower values of  $\vartheta^r$  many features will be disregarded and the final feature set will become rather small. Conversely, higher values of  $\vartheta^r$  close to one lead to low compression rates and large feature sets. We demand random access to each particular feature in the computed codebooks. Therefore, we translate the codebook into a feature vector  $\underline{f}^r$  of length  $T^r$  where  $T^r$  terms the number of codebook features. Let, as a shorthand,  $f_t^r$  address the feature with index  $t$  in the feature vector with index  $r$ , throughout.

In our case study, application of the vector quantization algorithm using feature calculator  $f^1$  with a similarity threshold of  $\vartheta^1 = 0.92$  yields the result presented in tab. 1. The final feature vector  $\underline{f}^1 = (f_t^1)_{1 \leq t \leq 8}$  comprises  $T^r = 8$  parquet graphs.





































## 5 Preselection Network

In this section we will present the second step of the proposed form of graph dynamics: a feedforward neural network that allows for preselection of salient learning examples, so-called *model candidates*, and likewise for preselection of *salient categories* the object in the presented image supposedly belongs to. This network will be called the *preselection network*. It is a fully-connected single layer perceptron [22] implementing a weighted majority voting scheme [10]. In the network’s input layer *position-invariant feature detectors* submit their assessments whether their reference feature is present in an image to dedicated input neurons while the output layer comprises one neuron for each predefined category. Synaptic weights are chosen such that the preselection network conforms to Linsker’s *infomax principle* [12], stating that the output of each cell preserves maximum information [25] about its input. This setup in conjunction with the application of the winner-take-most or winner-take-all nonlinearity as decision function implements a weighted majority voting scheme that allows for the desired preselection of salient categories and model candidates.

Here, the selection of salient categories and model candidates is only based on feature coincidences in image and model domain. As their spatial arrangement is disregarded false positives are frequent among the selected model candidates. To rule them out similar spatial arrangement of features will be asserted for the model to be selected in the correspondence-based verification part (Section 6).

### 5.1 Neural Model

In the preselection network we employ two types of generalized McCulloch & Pitts neurons [14], variant A with identity and variant B with a Heaviside threshold function  $H(\cdot)$  as output function. The output of a neuron of type A is equal to the weighted sum of its inputs  $\sum_{n=1}^N x_n w_n$  with  $x_n$  being the presynaptic neurons’ outputs and the  $w_n$  being synaptic weights. The output of a neuron of type B is 1, if the weighted sum of its inputs is greater than 0, and 0 otherwise.

CODEBOOK FEATURES	DISREGARDED FEATURES									
$f_1^1 =$  (1,1)	 (1,2) 0.96	 (1,3) 0.93	 (1,12) 0.97	 (1,13) 0.95	 (1,14) 0.92	 (1,22) 0.93	 (2,1) 0.95	 (2,12) 0.93		
$f_2^1 =$  (1,4)	 (1,5) 0.97	 (1,6) 0.97	 (1,7) 0.96	 (1,8) 0.94	 (1,15) 0.95	 (1,16) 0.93	 (2,2) 0.93	 (2,3) 0.94	 (2,4) 0.94	
	 (2,5) 0.93	 (2,6) 0.93	 (2,7) 0.93	 (2,8) 0.92	 (2,14) 0.92	 (2,15) 0.93	 (2,16) 0.93	 (2,17) 0.92	 (2,18) 0.92	
$f_3^1 =$  (1,9)	 (1,10) 0.95	 (1,19) 0.94	 (1,20) 0.96	 (2,9) 0.96	 (2,10) 0.94	 (2,19) 0.94	 (2,20) 0.95			
$f_4^1 =$  (1,11)	 (1,21) 0.94	 (1,31) 0.94	 (2,11) 0.97							
$f_5^1 =$  (1,17)	 (1,18) 0.97	 (1,24) 0.94	 (1,25) 0.93	 (1,26) 0.93	 (1,27) 0.95	 (1,28) 0.93				
$f_6^1 =$  (1,23)	 (1,29) 0.93	 (1,30) 0.92	 (2,22) 0.93	 (2,23) 0.94						
$f_7^1 =$  (2,13)	 (2,24) 0.93									
$f_8^1 =$  (2,21)	 (2,25) 0.96									

**Table 1: Case Study: Computation of Feature Vector  $\underline{f}^1$**  — The table shows the result of applying the vector quantization algorithm using feature calculator  $f^1$  with a similarity threshold of  $\vartheta^1 = 0.92$ . The table’s leftmost column comprises those parquet graphs that have been chosen as codebook features. The column on the right hand side shows the disregarded parquet graphs. The lower labels have been introduced in fig. 7, the upper labels are the similarities between the disregarded parquet graph and the respective codebook feature. The final feature vector  $\underline{f}^1 = (f_t^1)_{1 \leq t \leq 8}$  comprises  $T^r = 8$  parquet graphs.



## 5.2 Position-invariant Feature Detectors

To test the presence of a particular feature from the visual dictionary, in the following called *reference feature*, in an image we construct a *position-invariant feature detector* out of local feature detectors (Section 3.2). For this task, we distribute instances of local feature detectors uniformly over the image plane. For a given reference feature, combining the local feature detectors in a linear discriminant yields a position-invariant feature detector that returns 1 if the reference feature is observed at at least one position and 0 otherwise. Fig. 9 shows how a position-invariant feature detector is constructed for a feature  $f_t^r$  from the visual dictionary. For a given feature  $f_t^r$ , the symbol  $\tau_t^r$  denotes the respective position-invariant feature detector and  $\tau_t^r(I)$  (Eq. (7)) its result. We will say that a position-invariant feature detector  $\tau_t^r$  has *found* or *observed* its feature  $f_t^r$  in input image  $I$  if  $\tau_t^r(I) = 1$ . From now on, we use the term *feature detector* only for the position-invariant version.

$$\tau_t^r : \mathbb{I} \rightarrow \{0, 1\}; \tau_t^r(I) = H \left( \sum_{f \in f^r(I)} \varepsilon(f, f_t^r, \vartheta^r) \right) \quad (7)$$

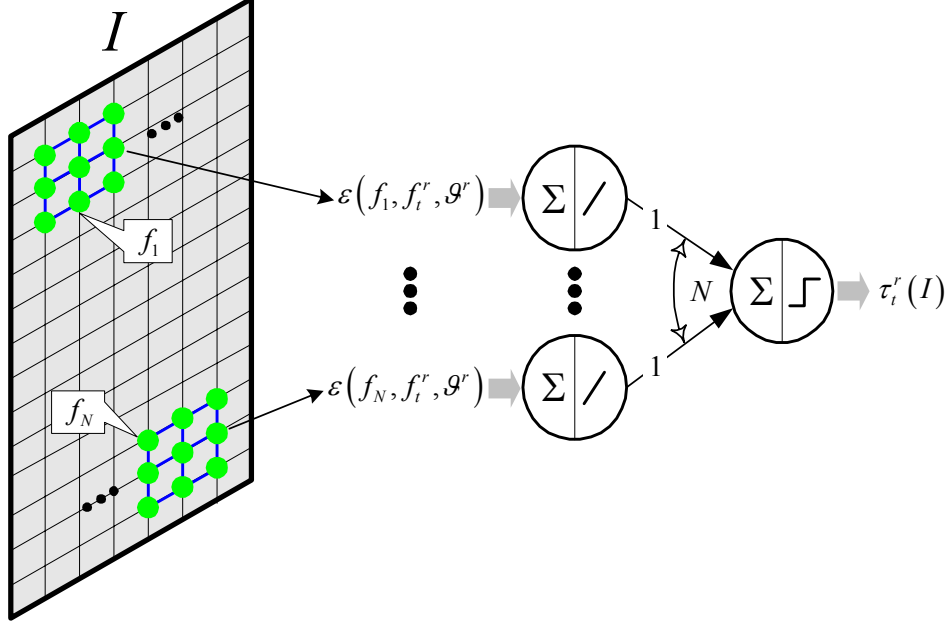
For the sake of simplicity we regard the feature detectors as the perceptron’s processing elements [22], rather than an additional layer.

Each time a feature detector has found its reference feature  $f_t^r$  in the input image, we add a pair  $(f, f_t^r)$  to a table of matching features, where  $f$  stems from the input image. That table is cleared before each image presentation.

$$\mathcal{F}_{match}(I) \leftarrow \mathcal{F}_{match}(I) \cup \{(f, f_t^r)\} \quad (8)$$

## 5.3 Weighting of Feature Detectors

From the example in tab. 1 it becomes clear that the feature detectors have varying relevance for the selection of salient categories. In the following the contributions of feature detectors to choosing salient categories are described through *measures of information*. Shannon has defined information as the decrease of uncertainty [25]. In this sense, a natural definition of the measures of information is presented in eq. (9). For a given feature detector  $\tau_t^r$  that has found its reference feature  $f_t^r$  in the input image and for a given partitioning  $\Pi^k$ , the information  $i_t^{r,k}$  that feature detector contributes to the decision about choosing categories of partitioning  $\Pi^k$  is defined by the difference between the largest possible amount of uncertainty and the feature detector’s amount of uncertainty encoded by the Shannon entropy  $\mathcal{H}_t^{r,k}$ .  $\mathcal{P}[\mathbb{C}_c^k | f_t^r]$  describes the conditional probability that the genuine category is



**Figure 9: Position-Invariant Feature Detector** — The position-Invariant feature detector returns 1 if a given feature  $f_t^r$  is present in image  $I$ , and 0 otherwise. At each grid position allowing for placement of a whole parquet graph a local feature detector is installed that compares the local graph with the reference feature  $f_t^r$ . Technically, this has been implemented by applying feature calculator  $f^r$  to the given image  $I$ . If the feature calculator returns a set of  $N$  parquet graphs  $\{f_n | 1 \leq n \leq N\}$ , each local feature detector compares its feature  $f_n$  with the reference feature  $f_t^r$  with respect to similarity threshold  $\vartheta^r$ . Then, each local feature detector passes its result into a single layer perceptron with  $N$  input units of type A, one output unit of type B, and feedforward connections of strength 1 between each input unit and the output neuron. The net's output is 1 if at least one of the local feature detectors has found its reference feature in the given image, and 0 otherwise. In this fashion, a position-invariant feature detector is instantiated for each feature in the visual dictionary.

$\mathbb{C}_c^k$  given that feature  $f_t^r$  has been observed. In this fashion measures of information are calculated for all features in the visual dictionary with respect to all partitionings of the learning set. Similar approaches are proposed in [31, 6].

$$i_t^{r,k} = \ln C^k - \mathcal{H}_t^{r,k} = \ln C^k + \sum_{c=1}^{C^k} \mathcal{P} [\mathbb{C}_c^k | f_t^r] \cdot \ln \mathcal{P} [\mathbb{C}_c^k | f_t^r] \quad (9)$$

For a given partitioning  $\Pi^k$ , the measures of information range between 0 and  $\ln C^k$ . If a feature occurs in all categories of that partitioning, the respective feature detector cannot make a contribution and, accordingly, its measure of information is 0. Conversely, if a feature occurs in only one category, the respective feature detector contributes maximally; its measure of information is  $\ln C^k$ .









Assuming that all prior probabilities for choosing a category are the same, the conditional probabilities  $\mathcal{P} [\mathbb{C}_c^k | f_t^r]$  are calculated through application of Bayes' rule (Eq. (10)). The  $n_t^r(\mathbb{C})$  denote the total number of observations of feature  $f_t^r$  in the images of the parameterized category  $\mathbb{C}$ . For a given category  $\mathbb{C}_c^k$  and a given feature  $f_t^r$ , we may interpret this probability as the frequency of that feature among the categories of partitioning  $\Pi^k$ . Tab. 2 demonstrates the calculation of measures of information in our case study.

$$\mathcal{P} [\mathbb{C}_c^k | f_t^r] = \frac{n_t^r(\mathbb{C}_c^k)}{\sum_{c'=1}^{C^k} n_t^r(\mathbb{C}_{c'}^k)} \quad (10)$$

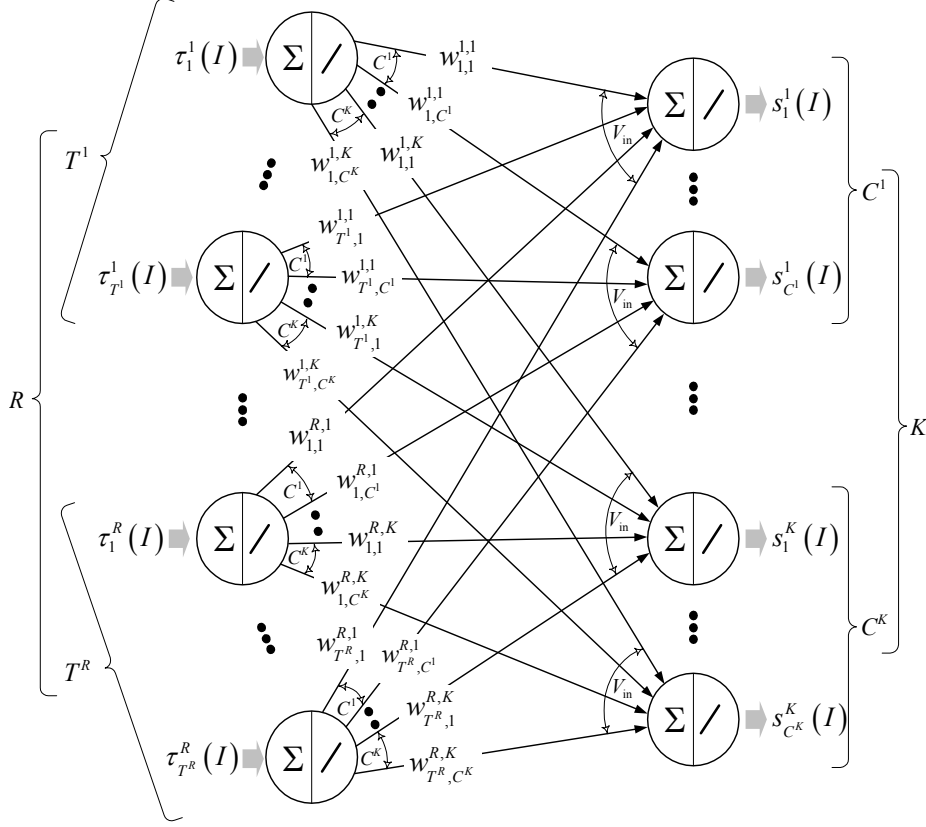
#### 5.4 Neurons, Connectivity, and Synaptic Weights

The preselection network is a single-layer perceptron comprising a layer of input and a layer of output neurons. In the network's input layer, we assign neurons of type A to the feature detectors. Thus, the network comprises  $V_{\text{in}} = \sum_{r=1}^R T^r$  input neurons. By definition, each input neuron passes the result of its feature detector into the network. In the network's output layer, we assign neurons of type A to the predefined categories. Accordingly, the network contains  $V_{\text{out}} = \sum_{k=1}^K C^k$  output neurons.

For fulfillment of the infomax principle, we define the synaptic weight  $w_{t,c}^{r,k}$  between the presynaptic neuron assigned to a feature detector  $\tau_t^r$  and the postsynaptic neuron assigned to a category  $\mathbb{C}_c^k$  as follows. Imagine that feature  $f_t^r$  can both be observed in the input image and in at least one image of that category. Then, this may be considered as a piece of evidence that the input image belongs to that category. Consequently, feature detector  $\tau_t^r$

FEATURE INDEX ( $t$ )	FEATURE ( $f_t^1$ )	$n_t^1(\mathbb{C}_1^1)$	$n_t^1(\mathbb{C}_2^1)$	$\mathcal{P}[\mathbb{C}_1^1 f_t^1]$	$\mathcal{P}[\mathbb{C}_2^1 f_t^1]$	$i_t^{1,1}$
1	 (1,1)	7	2	$\frac{7}{9}$	$\frac{2}{9}$	0.1634
2	 (1,4)	7	12	$\frac{7}{19}$	$\frac{12}{19}$	0.035
3	 (1,9)	4	4	$\frac{1}{2}$	$\frac{1}{2}$	0
4	 (1,11)	3	1	$\frac{3}{4}$	$\frac{1}{4}$	0.1307
5	 (1,17)	7	0	1	0	0.6931
6	 (1,23)	3	2	$\frac{3}{5}$	$\frac{2}{5}$	0.0201
7	 (2,13)	0	2	0	1	0.6931
8	 (2,21)	0	2	0	1	0.6931

**Table 2: Case Study: Calculation of Measures of Information** —  
The table demonstrates the calculation of the feature detectors’ measures of information. The parquet graphs in the second column stem from the visual dictionary and may be looked up in tab. 1. The number of feature occurrences  $n_t^1(\mathbb{C}_1^1)$  and  $n_t^1(\mathbb{C}_2^1)$  in columns three and four can be verified by counting the occurrences of the respective reference feature  $f_t^1$  within categories  $\mathbb{C}_1^1$  and  $\mathbb{C}_2^1$  (Tab. 1). The probabilities  $\mathcal{P}[\mathbb{C}_1^1|f_t^1]$  and  $\mathcal{P}[\mathbb{C}_2^1|f_t^1]$  in columns five and six have been calculated using eq. (10), and, finally, the measures of information  $i_t^{1,1}$  in column seven have been calculated using eq. (9). One can easily verify that the measures of information scale proportionally with the feature detectors’ contributions to choosing salient categories.



**Figure 10: Preselection Network** — The preselection network is a fully-connected single-layer perceptron. In its input layer neurons of type A have been assigned to the feature detectors. Accordingly, the network comprises  $V_{\text{in}} = \sum_{r=1}^R T^r$  input neurons. Each input neuron passes the binary result of its feature detector into the network. In the network's output layer neurons of type A have been assigned to the predefined categories. Accordingly, the network contains  $V_{\text{out}} = \sum_{k=1}^K C^k$  output neurons. The synaptic weights  $w_{t,c}^{r,k}$  are chosen in a way such that the whole network conforms to Linsker's infomax principle. The output of the postsynaptic neuron that has been assigned to a given category  $C_c^k$  will be called the saliency of that category and is denoted by  $s_c^k(I)$ .

$$\begin{aligned}
\underline{\underline{W}}^{1,1} &= \left( H \left( \sum_{I \in \mathbb{C}_c^1} \tau_t^1(I) \right) \cdot i_t^{1,1} \right)_{\substack{1 \leq c \leq 2 \\ 1 \leq t \leq 8}} \\
&= \begin{pmatrix} 0.1634 & 0.035 & 0 & 0.1307 & 0.6931 & 0.0201 & 0 & 0 \\ 0.1634 & 0.035 & 0 & 0.1307 & 0 & 0.0201 & 0.6931 & 0.6931 \end{pmatrix} \\
&=: \left( w_{t,c}^{1,1} \right)_{\substack{1 \leq c \leq 2 \\ 1 \leq t \leq 8}}
\end{aligned}$$

**Figure 11: Case Study: Weight Matrix** — In our case study, feature vector  $\underline{f}^1$  comprises eight features and the learning set has been partitioned into two categories. Accordingly, weight matrix  $\underline{\underline{W}}^{1,1}$  is of dimensions  $(2 \times 8)$ . The measures of information  $i_t^{1,1}$  that have been assigned to the respective feature detectors  $\tau_t^1$  can be looked up in tab. 2. Synaptic weights  $w_{7,1}^{1,1}$  and  $w_{8,1}^{1,1}$  have become 0 because the respective features  $f_7^1$  and  $f_8^1$  cannot be observed in the only image  $I_1$  of category  $\mathbb{C}_1^1$ . The same is likewise the case for weight  $w_{5,2}^{1,1}$ . The synaptic weights  $w_{3,1}^{1,1}$  and  $w_{3,2}^{1,1}$  have become 0 because the quantitative measure of information  $i_3^{1,1}$  is 0.

should contribute its quantitative amount of information  $i_t^{r,k}$  to the output of the postsynaptic neuron assigned to that category  $\mathbb{C}_c^k$ . Conversely, if that category contains only images in which that feature cannot be observed, the feature detector should never be allowed to make a contribution at all.

Using this construction rule of synaptic weights, we define  $R \cdot K$  matrices of synaptic weights  $\underline{\underline{W}}^{r,k}$ : one matrix per feature vector/partitioning combination. For a given feature vector  $\underline{f}^r$  and a given partitioning  $\Pi^k$ , weight matrix  $\underline{\underline{W}}^{r,k}$  (Eq. (11)) is of dimensions  $(C^k \times T^r)$ . That matrix comprises the synaptic weights  $w_{t,c}^{r,k}$  of the connections between the input neurons assigned to feature detectors  $\tau_t^r$  and the output neurons assigned to categories  $\mathbb{C}_c^k$ . The indices  $t$  of the presynaptic neurons range between 1 and  $T^r$  and the indices  $c$  of the postsynaptic neurons between 1 and  $C^k$ .

$$\underline{\underline{W}}^{r,k} = \left( H \left( \sum_{I' \in \mathbb{C}_c^k} \tau_t^r(I') \right) \cdot i_t^{r,k} \right)_{\substack{1 \leq c \leq C^k \\ 1 \leq t \leq T^r}} =: \left( w_{t,c}^{r,k} \right)_{\substack{1 \leq c \leq C^k \\ 1 \leq t \leq T^r}} \quad (11)$$

In our case study, feature vector  $\underline{f}^1$  comprises eight features and the learning set has been partitioned into two categories. Accordingly, weights matrix  $\underline{W}^{1,1}$  is of dimensions  $(2 \times 8)$ . The matrix is shown in fig. 11.

### 5.5 Saliencies

The output of the postsynaptic neuron of a category  $\mathbb{C}_c^k$  will be called the *saliency* of that category and is denoted by  $s_c^k(I)$ . With respect to an input image  $I$ , that saliency is defined as the sum of the measures of information  $i_t^{r,k}$  of those feature detectors  $\tau_t^r$  whose reference feature coincides in the input image and in at least one image of category  $\mathbb{C}_c^k$ . Thus, a saliency value is the accumulated evidence contributed by these feature detectors: the more pieces of evidence have been collected, the more likely the input image belongs to that category. For each partitioning of the learning set we can calculate a *saliency vector*  $\underline{s}^k$  of length  $C^k$  by summing up the matrix vector products of the weight matrices  $\underline{W}^{r,k}$  with the vector of feature detector responses  $(\tau_t^r(I))_{1 \leq t \leq T^r}$  over all feature  $R$  vectors in the visual dictionary (Eq. (12)). Fig. 10 shows the complete preselection network.

$$\underline{s}^k : \mathbb{I} \rightarrow \mathbb{R}^{C^k}; \quad \underline{s}^k(I) = \sum_{r=1}^R \underline{W}^{r,k} \cdot (\tau_t^r(I))_{1 \leq t \leq T^r} =: \left( s_c^k(I) \right)_{1 \leq c \leq C^k} \quad (12)$$

### 5.6 Selection of Salient Categories and Model Candidates

For selection of salient categories for the input image  $I$  we apply a winner-take-most nonlinearity as a decision rule. For a given partitioning  $\Pi^k$ , the set  $\Gamma^k(I)$  comprises all categories of the partitioning with superthreshold saliencies. The threshold is defined relative to the maximal saliency with a factor  $\theta^k$  with  $0 < \theta^k \leq 1$  (Eq. (13)). For  $\theta^k = 1$  only the most salient category will be selected: the decision rule becomes the winner-take-all nonlinearity then.

$$\Gamma^k(I) = \left\{ \mathbb{C}_c^k \in \Pi^k \mid s_c^k(I) \geq \theta^k \cdot \max_{1 \leq c' \leq C^k} \left\{ s_{c'}^k(I) \right\} \right\} \quad (13)$$

A necessary requirement for a learning image to become a model for the given input image is to be a member of each salient category. Accordingly, the set of model candidates  $\mathbb{M}(I)$  for input image  $I$  is the set of those learning images that belong to all salient categories (Eq. (14)). Learning images that satisfy this requirement are called *model candidates*. They will be passed to the correspondence-based verification for further selection.

$$\mathbb{M}(I) = \bigcap_{k=1}^K \bigcup_{\mathbb{C} \in \Gamma^k(I)} \mathbb{C} \quad (14)$$

## 6 Verification of Model Candidates

Up to here, model candidates have been selected by set intersection on salient categories (Eq. (14)). The categories' saliencies as computed by the preselection network are solely based on the detection of coincidental features. The spatial arrangement of features, parquet graphs in our case, has been fully ignored, which can be particularly harmful in cases of multiple objects or structured backgrounds.

In the following model candidates are further verified through asserting that the features be in similar spatial arrangement for the model to be selected. More specifically, they are verified with a variant of elastic graph matching [33, 9, 40]. For each model candidate an image and a model graph are dynamically constructed through assembling corresponding features into larger graphs according to their spatial arrangement. For each model candidate the similarity between its image and model graph is computed. The model candidate that attains the best similarity is chosen as the model for the input image. Its model graph is the closest possible representation of the object in the input image with respect to the learning set.

### 6.1 Construction of Graphs

Construction of graphs proceeds in three steps. First, from the table of matching features, which has been assembled while computing the categories' saliencies (Eq. (8)), all feature pairs whose model feature can be observed in the current model candidate are collected in a smaller table of corresponding features. Second, templates of an image and of a model graph are instantiated with unlabeled nodes. The number and the positioning of nodes is determined by the valid nodes of image and model parquet graphs. Third, at each node position, separately for image and model graph, a bunch of Gabor jets is assembled whose jets stem from valid-labeled parquet graph nodes located at that position. The respective nodes of the image or model graph become attributed with these bunches.

#### 6.1.1 Table of Corresponding Features

During calculation of the categories' saliencies pairs of matching features, one feature from the input image and the other from a learning image, have been collected in a table of matching features (Eq. (8)). Given a model candidate  $M \in \mathbb{M}(I)$  for an input image  $I$  (Section 5.6), all feature pairs whose model feature can be observed in the current model candidate are transferred into a table of *corresponding* features  $\mathcal{F}_{corr}(I, M)$  (Eq. (15)).



That table is used for efficient aggregation of local features, parquet graphs in our case, into larger model and image graphs. We assume that the table of corresponding features comprises  $N$  feature pairs, a number which depends implicitly on the model candidate. Let  $f_n^I$  denote the  $n$ -th image and let  $f_n^M$  denote the  $n$ -th model parquet graph. Note that from now on we speak of *corresponding* rather than of *matching* parquet graphs and assume that those graphs establish a field of contiguous point-to-point correspondences between the input image and the model candidate.

$$\mathcal{F}_{corr}(I, M) = \left\{ (f_n^I, f_n^M) \in \mathcal{F}_{match}(I) \mid 1 \leq n \leq N \wedge H \left( \sum_{r=1}^R \sum_{f \in f^r(M)} \varepsilon(f, f_n^M, 1) \right) = 1 \right\} \quad (15)$$

Nodes of parquet graphs are attributed with a triple consisting of an absolute image position, a Gabor jet derived from an image at that position, and a validity flag. In order to be able to globally address node label components the following notation is introduced: the nodes of image parquet graphs are attributed with triples  $(\underline{x}_{n,v}^I, \mathcal{J}_{n,v}^I, b_{n,v}^I)$  where  $n$  specifies the index of the feature pair in the table of corresponding features and  $v$  encodes the node index. The same notation is used for model parquet graphs, with an superscript  $M$  for distinction (Eq. (16)).

$$\begin{aligned} f_n^I &= \{ (\underline{x}_{n,v}^I, \mathcal{J}_{n,v}^I, b_{n,v}^I) \mid 1 \leq v \leq V \} \\ f_n^M &= \{ (\underline{x}_{n,v}^M, \mathcal{J}_{n,v}^M, b_{n,v}^M) \mid 1 \leq v \leq V \} \end{aligned} \quad (16)$$

### 6.1.2 Graph Templates

First templates of an image and of a model graph are instantiated without node labels. Number and positioning of nodes are determined by the valid nodes of image and model parquet graphs. For this purpose, the positions of all valid nodes of image and model parquet graphs are collected into sets  $\mathbb{X}^I$  and  $\mathbb{X}^M$ , respectively (Eq. (17)). The creation of graph templates is illustrated in fig. 12.

$$\begin{aligned} \mathbb{X}^I &= \bigcup_{n,v} \{ \underline{x}_{n,v}^I \mid b_{n,v}^I = 1 \} \\ \mathbb{X}^M &= \bigcup_{n,v} \{ \underline{x}_{n,v}^M \mid b_{n,v}^M = 1 \} \end{aligned} \quad (17)$$

### 6.1.3 Node Labels

The nodes of model and image graphs become attributed with bunches of Gabor jets: nodes of image graphs become labeled with bunches of Gabor

jets that stem from valid-labeled nodes of image parquet graphs located at a given position  $\underline{x}$  in the input image. Nodes of model graphs are just the same attributed with bunches of jets that stem from valid-labeled nodes of model parquet graphs located at a given position  $\underline{x}$  in the model candidate. The assembly of Gabor jets derived from image and model parquet graphs into bunches is also illustrated in fig. 12.

$$\begin{aligned}\beta^I(\underline{x}) &= \bigcup_{n,v} \{ \mathcal{J}_{n,v}^I \mid \underline{x}_{n,v}^I = \underline{x} \wedge b_{n,v}^I = 1 \} \\ \beta^M(\underline{x}) &= \bigcup_{n,v} \{ \mathcal{J}_{n,v}^M \mid \underline{x}_{n,v}^M = \underline{x} \wedge b_{n,v}^M = 1 \}\end{aligned}\tag{18}$$

For the assessment whether a point in the input image defined by a node of the image graph corresponds to a point in the model candidate defined by a node of the model graph a measure of similarity between two bunches of jets is needed. The similarity between two bunches is computed with a cross run over all their jets (Eq. (19)). The function returns the maximal similarity between jets computed within that cross run. The measure of similarity between two jets is based on the Gabor amplitudes (Eq. (3)). If one of the bunches is empty the similarity between them yields 0.

$$s_{bunch}(\beta, \beta') = \begin{cases} 0 & \text{if } \beta = \emptyset \vee \beta' = \emptyset \\ \max_{\mathcal{J} \in \beta, \mathcal{J}' \in \beta'} \{s_{abs}(\mathcal{J}, \mathcal{J}')\} & \text{otherwise} \end{cases}\tag{19}$$

#### 6.1.4 Graphs

Like parquet graphs, image and model graphs are specified by a set of node labels. Node labels comprise an absolute position in the input or model image drawn from the sets of node positions (Eq. (17)) and the bunch that has been assembled at that position (Eq. (18)). The image graph is decorated with an superscript  $I$  while the model graph receives an superscript  $M$  (Eq. (20)).

$$\begin{aligned}\mathcal{G}^I &= \bigcup_{\underline{x} \in \mathbb{X}^I} \{ (\underline{x}, \beta^I(\underline{x})) \} \\ \mathcal{G}^M &= \bigcup_{\underline{x} \in \mathbb{X}^M} \{ (\underline{x}, \beta^M(\underline{x})) \}\end{aligned}\tag{20}$$

Model graphs of well-suited model candidates provide an approximation of the object contained in the input image by features present in the visual dictionary. Fig. 2 shows a number of model graphs (third column) that have been constructed for the input image given in the first column. The reconstructions from the model graphs of the first two model candidates in column four demonstrate that the emerged model graphs describe the object contained in the input image well.

The constructed graphs are to some extent reminiscent of bunch graphs [39, 40]. Nevertheless, since they are restricted to only one model candidate we rather speak of model instead of bunch graphs. It is however worthwhile mentioning that the presented procedure of graph construction may as well serve for creation of bunch graphs. For compilation of bunch graphs the set of corresponding features has to provide feature pairs of model candidates picked from a carefully chosen subset  $\tilde{\mathbb{M}}(I)$  of the set of model candidates (Eq. (21)). The graph construction procedure is then as well applicable to the construction of bunch graphs.

$$\mathcal{F}_{corr}^{bunch}(I, \tilde{\mathbb{M}}(I)) = \bigcup_{M \in \tilde{\mathbb{M}}(I)} \mathcal{F}_{corr}(I, M) \quad (21)$$

## 6.2 Matching

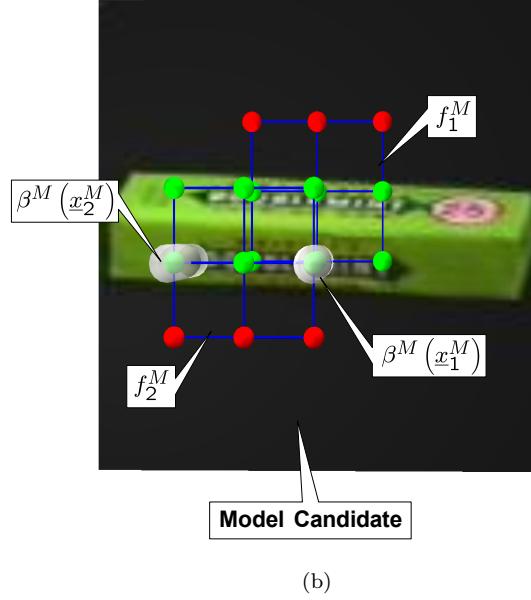
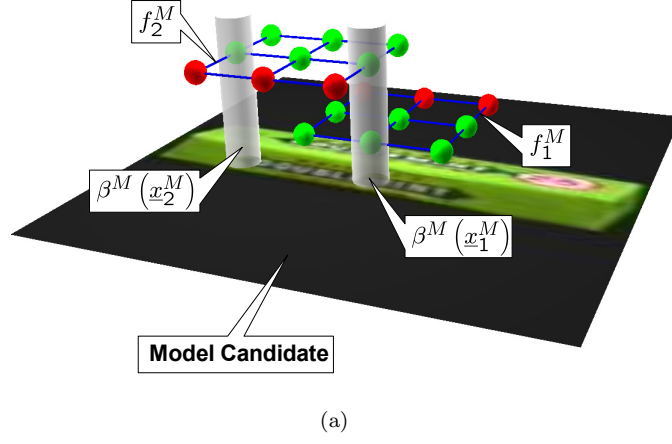
In order to find the object contained in the input image and to overcome accidentally corresponding parquet graphs, the emerged model graph is moved as a template over the entire image plane. This action can be compared with the *scan global move* which is usually performed as the first step of elastic graph matching [9, 40]. It is also very similar to multidimensional template matching [42]. For each translation of the model graph the similarity between model and image graph is computed. The translation vector that yields the best similarity defines the optimal placement of the model graph in the image plane. In the process, the model graph's absolute node positions are transformed into relative ones by subtracting a displacement vector  $\underline{t}_0$  from the positions of the model graph's nodes. That vector is chosen such that after subtraction the smallest  $x$  and the smallest  $y$  coordinate become zero (Eq. (22)). However, the  $y$  coordinate of the leftmost node is not necessarily 0. The same is the case for the  $x$  coordinate of the uppermost node.

$$\underline{t}_0 = \left( \min_{n,v} \left\{ (x_{n,v}^M)_x \right\}, \min_{n,v} \left\{ (x_{n,v}^M)_y \right\} \right)^\top \quad (22)$$

The similarity between model and image graph with respect to a given translation vector  $\underline{t}$  is defined as the normalized sum of the similarities between the image and model bunches (Eq. (23)).

$$s(I, M, \underline{t}) = |\mathcal{G}^M|^{-1} \cdot \sum_{(\underline{x}^M, \beta^M) \in \mathcal{G}^M} s_{bunch}(\beta^I(\underline{x}^M - \underline{t}_0 + \underline{t}), \beta^M) \quad (23)$$

In order to find the object contained in the input image the model graph is iteratively translated about a displacement vector in the image plane so that the measure of similarity between model and image graph becomes



**Figure 12: Construction of Model Graphs** — *Fig. (a) provides a side, fig. (b) a top view of the same setup. For clarity, both figures show only two overlapping model parquet graphs  $f_1^M$  and  $f_2^M$  drawn from the table of corresponding features. For illustration of the overlap the graphs are drawn in a stacked manner. Number and position of the model graph's nodes are determined by the valid-labeled model parquet graph nodes (green nodes). Nodes that reside in the background have been marked as invalid (red nodes). In fig. (b) the shape of the emerging model graph can be seen. Compilation of bunches is demonstrated with two bunches only. Like stringing pearls, all valid Gabor jets at position  $\underline{x}_1^M$  are collected into bunch  $\beta^M(\underline{x}_1^M)$  and those at positions  $\underline{x}_2^M$  become assembled into bunch  $\beta^M(\underline{x}_2^M)$ . From fig. (a) we can learn that bunch  $\beta^M(\underline{x}_1^M)$  comprises two jets while bunch  $\beta^M(\underline{x}_2^M)$  contains only one jet. Image graphs are constructed in the very same fashion.*

maximal (Eq. (24)). The model graph thus moves to the position where the object contained in the input image is located. Let  $s_{best}(I, M)$  denote the similarity attained at that position. The displacement vectors  $\underline{t}$  stem from a set  $\mathbb{G}$  of all grid points defined by the given inter-node distances  $\Delta x$  and  $\Delta y$  between two consecutive nodes in the parquet graphs (Section 4).

$$s_{best}(I, M) = \max_{\underline{t} \in \mathbb{G}} \left\{ s(I, M, \underline{t}) \right\} \quad (24)$$

### 6.3 Model Selection

For selection of the model, the most similar learning image for the given input image, an image and a model graph are constructed for each model candidate. The model candidate that attains the best similarity between its model and image graph is chosen as the model for the input image (Eq. (25)).

$$M_{best} = \arg \max_{M \in \mathbb{M}(I)} \left\{ s_{best}(I, M) \right\} \quad (25)$$

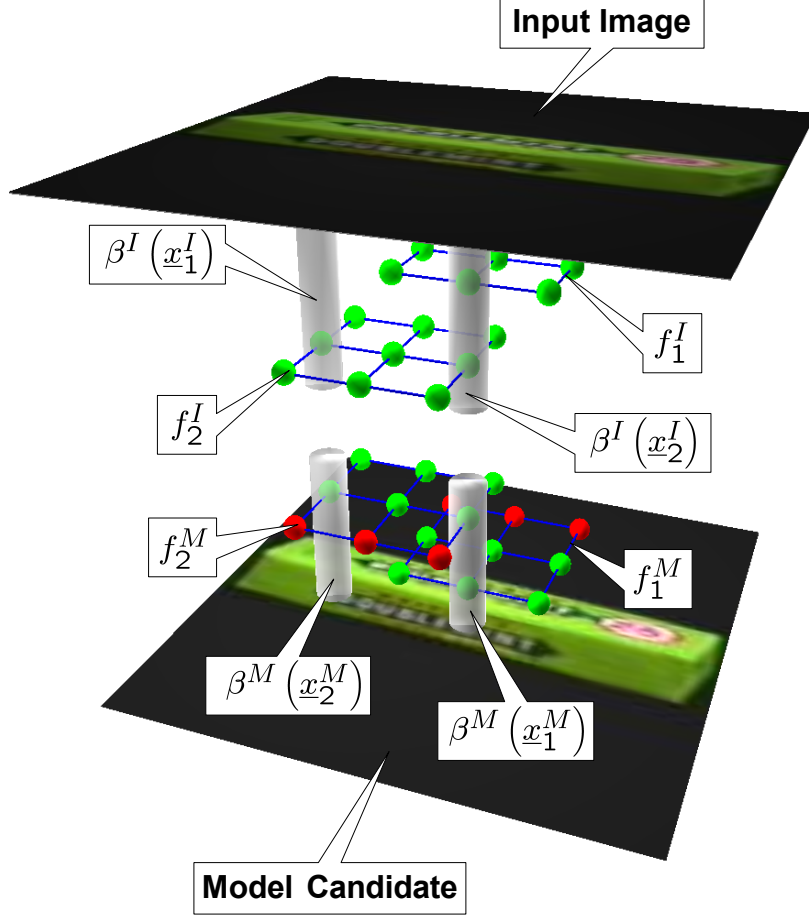
In fig. 2 four model candidates (column two) have been computed for the given input image (column one). The similarities attained through matching image against model graphs are annotated to the reconstructions from the model graphs (column four). Since the first model candidate yields the highest similarity, it is chosen as the model for the object in the input image.

## 7 Experiments

We report experimental results derived from standard databases for object recognition and categorization. The results are excerpted from [37].

### 7.1 Object Recognition

Object recognition experiments were conducted on the COIL-100 image database [17]. That database contains images of 100 objects in 72 poses per object, thus, 7200 image in total. We present the results of three experiments. First, we investigated the recognition performance with respect to object identity and pose for input images containing a single object, second, we analyzed the recognition performance for input images containing multiple objects, and third, recognition performance was measured for images of partially occluded objects.



**Figure 13: Matching Setup** — The setup consists of the input image, the model candidate, and the graphs constructed using the method presented in section 6.1. For clarity only two pairs of corresponding parquet graphs have been taken from the table of corresponding features. Parquet graph  $f_1^I$  corresponds to  $f_1^M$  and that  $f_2^I$  corresponds to  $f_2^M$ . Like in fig. 12, green nodes represent nodes that have been marked as valid and red nodes represent nodes that have been marked as invalid due to residing in the background. Since only learning images provide figure-ground information, invalid nodes appear only in the model parquet graphs. The compilation of bunches is illustrated for two exemplary positions  $\underline{x}_1^I$  and  $\underline{x}_2^I$  in the input image and  $\underline{x}_1^M$  and  $\underline{x}_2^M$  in the model candidate. In order to find the object in the input image the model graph is iteratively moved over the entire image plane and matched to the image graph.

Experimental results were attained in a fivefold cross-validation [41]. We thus created five pairs of disjoint learning and testing sets from all COIL-100 images. The learning sets comprise 56, the testing sets 14 views per object, thus, 5600 or 1400 images in total, respectively. The object recognition application is designed to simultaneously recognize the object’s identity and pose. This is achieved by creating  $K = 1$  partitioning of the learning set. That partitioning consists of single-element categories. Moreover, from each learning set a visual dictionary with  $R = 2$  feature vectors of increasing length was calculated using similarity thresholds of  $\vartheta^1 = 0.9$  and  $\vartheta^2 = 0.95$  (Algorithm 1). Sorting feature vectors according to detailedness is harnessed in a procedure that allows for accelerated search of features [37]. Computation and parameters of the Gabor features are the same as in [9, 40], i.e., five scales, eight orientations,  $k_{max} = \frac{\pi}{2}$ ,  $k_{step} = \sqrt{2}$ , and  $\sigma = 2\pi$ . For this parameterization, the horizontal and vertical node distances  $\Delta x$  and  $\Delta y$  are set to 10 pixels.

In the following we present recognition results computed within the cross-validation and their dependence on relative weighting of the feature- and correspondence-based parts. Each data point was averaged over  $5 \cdot 1400 = 7000$  single measurements. Weighting of the feature- and correspondence-based part is controlled by the threshold scaling factor  $\theta^1$  (Eq. (13)) that ranges between 0.1 and 1, sampled in 0.1-steps.  $\theta^1$  determines the final number of model candidates that are passed to the correspondence-based verification part. For  $\theta^1 = 1$  only one model candidates is selected while for low values the set of model candidates encompasses large portions of the learning set. That factor thus enables us to adjust the balance between the feature- and correspondence-based parts.

### 7.1.1 Recognition of Single Objects

In the first experiment we present images containing a single object and evaluate the recognition performance with respect to object identity and pose. We analyzed the system’s performance for each of the combinations segmented/unsegmented images and preselection network conforming/non-conforming to the infomax principle (Section 5). The experiment is subdivided into eight test cases. In the first four test cases the recognition performance with respect to object identity was evaluated for each of these combinations while the system’s ability to recognize the objects’ poses was investigated in the remaining four test cases. Since the images of the COIL-100 database are perfectly segmented, the unsegmented images have been manually created by pasting the object into a cluttered background consisting of arbitrarily chosen image patches of random size derived from the other test images of the current testing set. Fig. 14 shows an example of



**Figure 14: Input Images of a Single Object** — *The figure shows an object from the COIL-100 database [17] as (a) segmented and (b) unsegmented image. Since the images of that database are perfectly segmented, the unsegmented images have been manually created through pasting the object contained in the segmented image into a cluttered background consisting of arbitrarily chosen image patches of random size derived from the other test images of the current testing set.*

a segmented and of an unsegmented image. In order to assess the usefulness of the choice of synaptic weights according to eq. (11) the preselection networks are made incompatible to the infomax principle by putting their weights out of tune by eq. (26). Choosing the synaptic weights in this fashion the categories' saliencies become simple counters of feature coincidences, the weighted majority voting scheme degenerates to a non-weighted one.

$$\underline{\underline{\hat{W}}}^{r,k} = \left( H \left( \sum_{I' \in \mathbb{C}_c^k} \tau_t^r(I') \right) \right)_{\substack{1 \leq c \leq C^k \\ 1 \leq t \leq T^r}} =: \left( \hat{w}_{t,c}^{r,k} \right)_{\substack{1 \leq c \leq C^k \\ 1 \leq t \leq T^r}} \quad (26)$$

The recognition performance with respect to object identity is shown in fig. 15 (a). We considered the object in the test image to be correctly recognized if test and model image show the same object regardless of its pose. Throughout, better recognition rates are attained if segmented images were presented. Moreover, the infomax principle always slightly improved performance where that improvement is continually exhausted in gradually putting more and more emphasis on the correspondence-based part, i.e., the achieved improvement is continually used up while moving from the left to the right hand side in fig. 15. Most interestingly, a well-balanced combination of the feature and correspondence-based part led to optimal performance, throughout. Only for such well-balanced combinations the selection of model candidates is optimally carried out in the sense that neither too few nor too many learning images become chosen as model candidates.

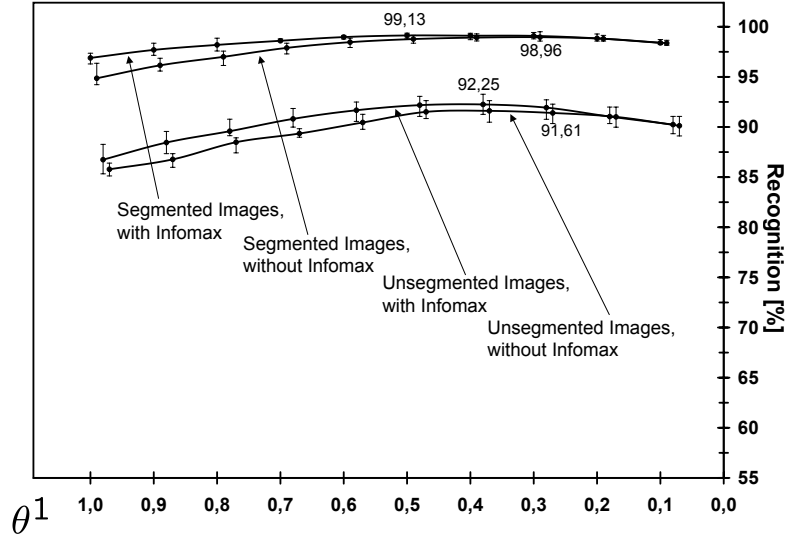


If the number of model candidates is too small, the spectrum of alternatives the correspondence-based part can choose from becomes too limited. This is especially harmful, if false positives are frequent among model candidates. Conversely, the number of false positives among model candidates unavoidably increases with overemphasis of the correspondence-based part: for too low values of the threshold scaling factor even learning images of weakly salient categories become selected as model candidates. Accordingly, the probability of choosing a false positive as the final model increases, the average recognition rate decreases. The same findings hold true for the performance with respect to object pose given in fig. 15 (b). The average pose errors were calculated over the absolute values of angle differences of correctly recognized, non-rotation-symmetric objects. Note that two consecutive learning images of the same object are at least five degrees apart.

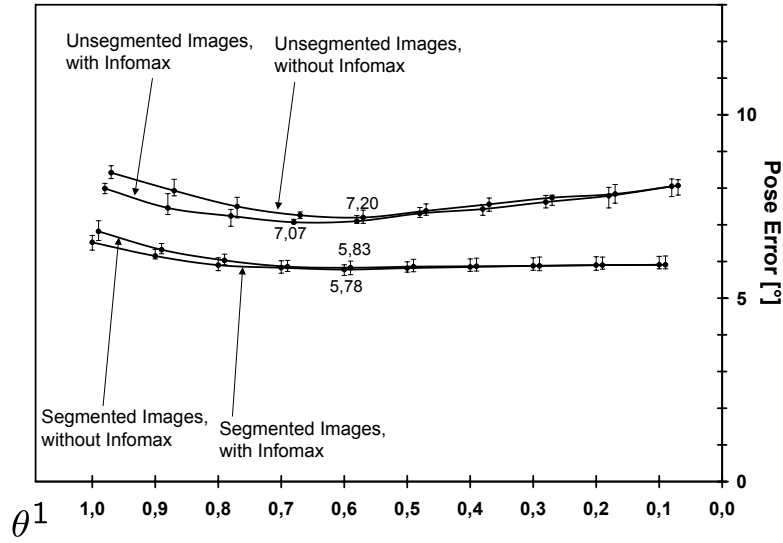
### 7.1.2 Recognition of Multiple Objects

The second experiment is concerned with the recognition of multiple, simultaneously presented, non-overlapping objects with respect to the same weightings of the feature-based and correspondence-based part as in the first experiment. In the current experiment only the recognition performance with respect to object identity was evaluated. The experiment is subdivided into six test cases. In the first three test cases we simultaneously presented  $N$  ( $N = 2, 3, 4$ ) objects placed in front of a plain black background while in the last three test cases cluttered background was manually added. The procedure of background construction was the same as in the first experiment. Fig. 16 shows two images containing four objects with and without background. Objects are randomly picked, a test image contains only different ones, and each object appears at least once. The system in this case returns the  $N$  most similar models. Each coincidence with one of the presented objects is accounted as a successful recognition response. The average recognition rates in fig. 17 were calculated over all responses.

From the results presented in fig. 17 we learn that, compared to the single-object experiments, the point of optimal recognition performance has considerably moved to the right hand side. Putting more emphasis on the correspondence-based verification part thus improves recognition performance. However, overemphasis of that part caused by too small values of the threshold scaling factor  $\theta^1$  again led to a decrease in recognition performance. This phenomenon can be observed in the test cases with unsegmented images (Fig. 17 (b)). Moreover, presentation of segmented images yielded better results. For segmented (Fig. 17 (a)) and unsegmented images (Fig. 17 (b)) the system's performance smoothly degraded with the number of simultaneously presented objects.



(a)



(b)

**Figure 15: Recognition of Single Objects** — The figure shows the recognition performance with respect to (a) object identity and (b) object pose depending on relative weighting of the feature- and correspondence-based parts controlled by  $\theta^1$ . This parameter determines the final number of model candidates that are passed to the correspondence-based verification part. The best results are annotated to the respective data points. The results were better for segmented images. Optimal performance was attained by satisfying the infomax principle and a well-balanced combination of the feature- and correspondence-based parts.



(a)



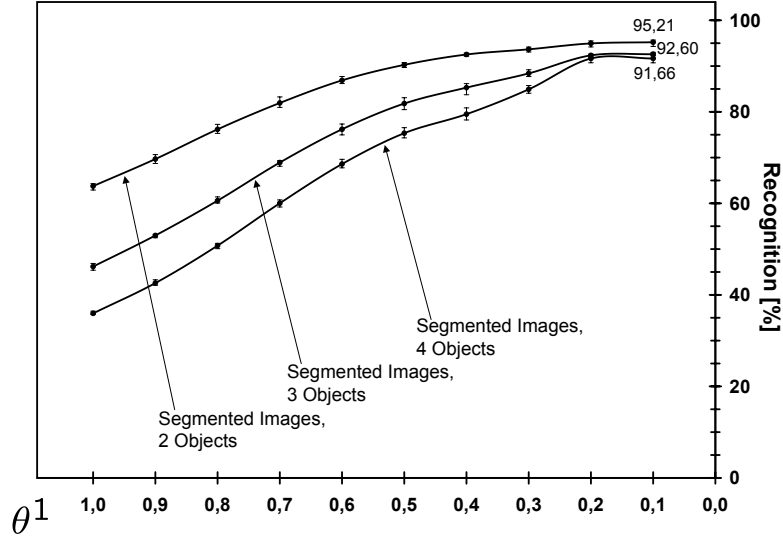
(b)

**Figure 16: Input Images of Multiple Objects** — *The figure shows an example of (a) a segmented and (b) an unsegmented input image containing four objects drawn from the COIL-100 database [17]. Backgrounds were constructed in the same fashion as in the first experiment.*

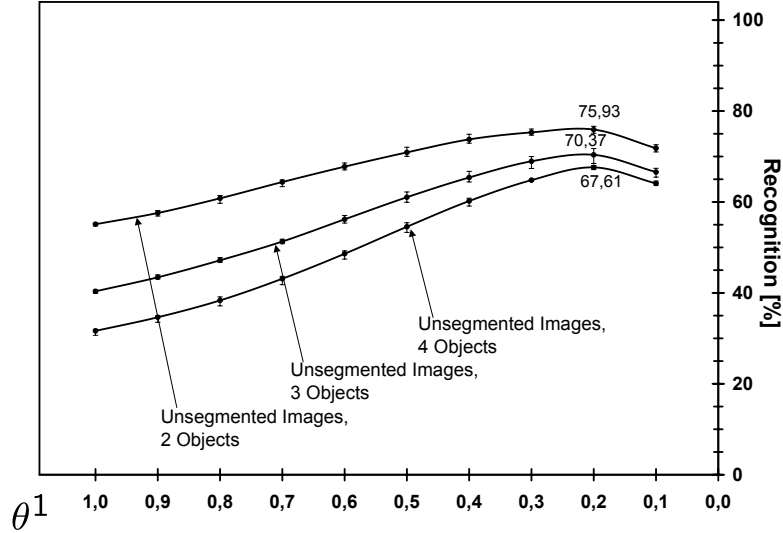
### 7.1.3 Recognition of Partially Occluded Objects

While in the second experiment the objects were presented in a non-overlapping manner, the third and last experiment is concerned with the recognition of partially occluded objects with respect to the same weightings of the feature-based and correspondence-based part as in the first two experiments. Again, we only evaluated recognition performance with respect to object identity. The experiment is subdivided into twelve test cases. In the first six test cases we simultaneously presented two objects where 0-50% of the object on the left was occluded by the object on the right. Occluded and occluding objects were different and randomly picked, each object appears at least once as occluded. In the last six test cases cluttered background was added. The procedure of background construction and accounting of recognition responses was the same as in the second experiment. Fig. 18 shows input images of partially occluded objects, fig. 19 the average recognition rates.

Like in the second experiment, we learn from the results presented in fig. 19 that emphasis of the correspondence-based part improves recognition perfor-

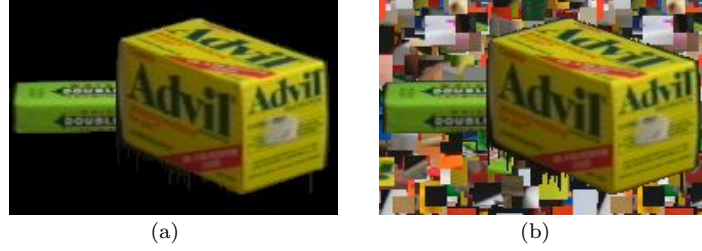


(a)



(b)

**Figure 17: Recognition of Multiple Objects** — The figure shows the recognition performance with respect to object identity in the case of multiple non-overlapping object; (a) for segmented, (b) for unsegmented images. Compared to the first experiment, the point of optimal recognition performance has considerably moved to the right: correspondence-based verification is more important in the case of multiple objects, overemphasis, however, led to a decline. Presentation of segmented images yielded better results. Performance smoothly degraded with the number of simultaneously presented objects.

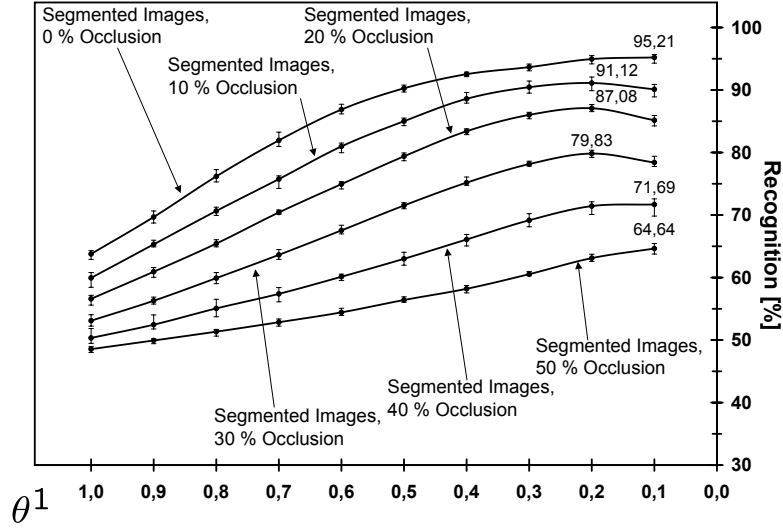


**Figure 18: Input Images of Partially Occluded Object** — *The figure shows (a) a segmented and (b) an unsegmented input image of partially occluded objects. The procedure of background construction was the same as in the first experiment. In this example, the occluding object covers about fifty percent of the occluded object.*

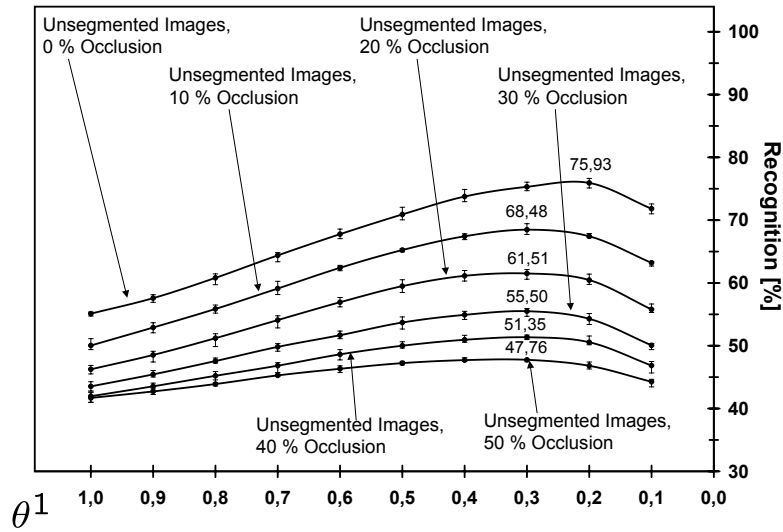
mance. Again, overemphasis of that part however led to a decline of recognition performance. Moreover, presentation of segmented images yielded better results. For segmented (Fig. 17 (a)) and unsegmented images (Fig. 17 (b)) the system’s performance smoothly degrades with the amount of occlusion.

## 7.2 Object Categorization

Object categorization experiments were conducted on the ETH-80 image database [11]. That database contains images of eight categories namely apples, pears, tomatoes, dogs, horses, cows, cups, and cars of ten identities per category and 41 images in different poses per identity. The database thus consists of 3280 images in total. An interesting question with respect to object categorization is whether a given hierarchical organization of categories can be harnessed to improve categorization performance; the question how such a hierarchical organization is set up is however not addressed here. We present the results of two experiments. First, we evaluated categorization performance if the decision about the final category the object in the input image is supposed to belong to relies on a given hierarchical organization of categories. We employ the hierarchy given in fig. 4. Second, we evaluated categorization performance if no such hierarchy is given.



(a)



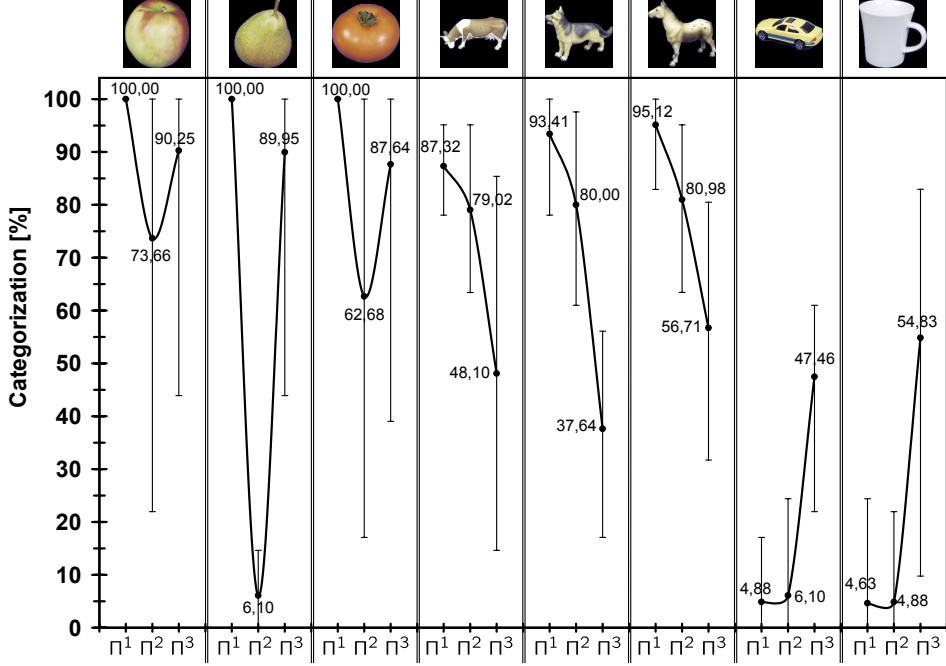
(b)

**Figure 19: Recognition of Partially Occluded Objects** — The figure shows the recognition performance with respect to object identity in the case of partially occluded objects, (a) for segmented, (b) for unsegmented images. Like in the second experiment, emphasis of the correspondence-based verification part improves performance, overemphasis of that part however led to a decline. Presentation of segmented images yielded better results. The system's performance smoothly degraded with the amount of occlusion.

### 7.2.1 Categorization of Objects Using Hierarchically Organized Categories

Results of the first experiment were attained in a leave-one-object-out cross-validation [41]. This means that the system was trained with the images of 79 objects and tested with the images of one unknown object. We thus created 80 pairs of learning and testing sets. The learning sets contained 3239, the testing sets 41 images. We hierarchically organized the images into categories of  $K = 3$  partitionings as given in fig. 4. The threshold scaling factors  $\theta^k$  for selection of salient categories of partitionings  $\Pi^k$  with  $k \in \{1, 2, 3\}$  were all set to 0.4 (Eq. (13)). The parameterization of parquet graph features was the same as in the object recognition experiments. For  $\Pi^1$  and  $\Pi^2$  we considered an object to be correctly categorized if exactly one category out of these was selected as salient and the presented object belongs to that category. For  $\Pi^3$  a set of model candidates was calculated by intersection of salient categories (Eq. (14)). The model candidates of that set were passed to the correspondence-based verification part. We considered the presented object to be correctly categorized if it belonged to the same category as the object in the model image.

Fig. 20 displays the averaged categorization rates computed within the leave-one-object-out cross-validation broken down into the original eight categories of apples, pears, tomatoes, dogs, horses, cows, cups, and cars. Each data point was averaged over  $10 \cdot 41 = 410$  single measurements. Generally, one finds that categorization performance depends considerably on the sampling of categories. In this sense the system categorizes apples, pears, and tomatoes well but obviously experiences difficulties in categorizing cows, dogs, horses, cars, and cups; the intra-category variations among the identities within these categories are too large. It is thus reasonable to assume that categorization performance may be improved by adding more learning examples to those categories. Moreover, the feature-based part's ability to unambiguously assign the object contained in the input image to the meta-categories of partitioning  $\Pi^1$  and  $\Pi^2$  is obviously limited. This deficiency is especially prominent in the results attained for the categorization of pears, cars and cups. Due to the imbalance between natural and man-made objects, the attained results for cars and cups are even worse than those for pears. The correspondence-based verification part can to some extent compensate for this shortcoming and greatly improves categorization performance for apples, pears, tomatoes, cars, and cups. However, the shortage of learning examples especially in the animal categories can only be fixed by additional training images.



**Figure 20: Categorization of Objects Using Hierarchically Organized Categories** — The averaged categorization rates computed within the leave-one-object-out cross-validation are displayed. Each data point was averaged over 410 single measurements. Categorization performance depends considerably on the sampling of categories. The feature-based part’s ability to unambiguously assign the object in the input image to the categories of partitionings  $\Pi^1$  and  $\Pi^2$  is obviously limited. For most cases, the correspondence-based verification part can compensate for this shortcoming, but not for the shortage of learning examples, especially in the animal categories.

### 7.2.2 Categorization of Objects Using Single-Element Categories

For evaluation of the system’s performance without predefined hierarchical organization of categories we arranged the learning set into  $K = 1$  partitioning of single-element categories. We considered the object in the input image to be correctly categorized if it belonged to the same original category of apples, pears, tomatoes, cows, dogs, horses, cars, or cups as the object in the model image. The attained results depending on  $\theta^1$  are given in fig. 21; for clarity the curves are distributed over two subfigures. All other parameters were the same as above.



As in the object recognition experiments, a well-balanced combination of the feature-based and the correspondence-based part allows for optimal categorization performance. The expectation that categorization performance would benefit from hierarchical organization of categories could not be substantiated. In the case of apples, tomatoes, cows, horses, cars, and cups average categorization performance was considerably better without hierarchy; only for pears and dogs categorization could benefit slightly.

The categorization rates are below or close to those presented in [11]. That object categorization system, however, integrates color, texture, and shape features while our system only relies on local texture information. At least the feature-based part of the technique described in this paper can work with any convenient feature type [38]. One can thus expect to further improve categorization performance if more feature types become incorporated.

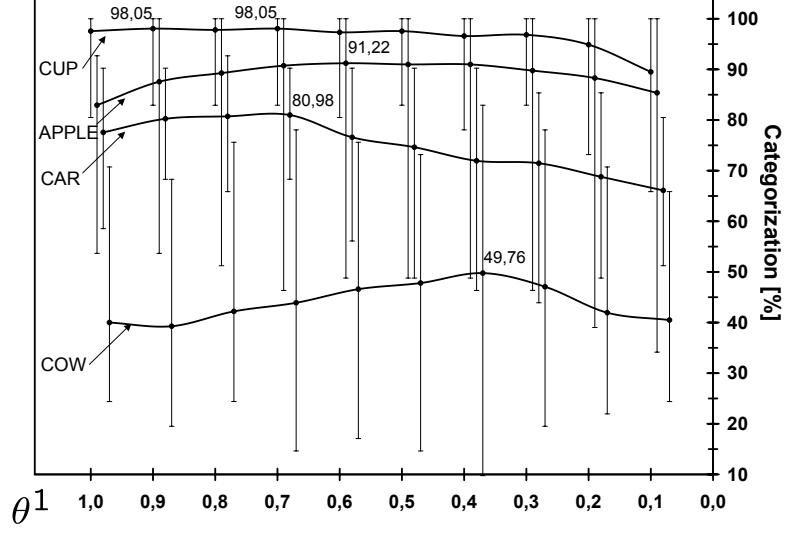
## 8 Discussion

We have presented an algorithm that employs a combination of rapid feature-based preselection with self-organized model graph creation and subsequent correspondence-based verification of model candidates. This hybrid method outperformed both purely feature-based and purely correspondence-based approaches.

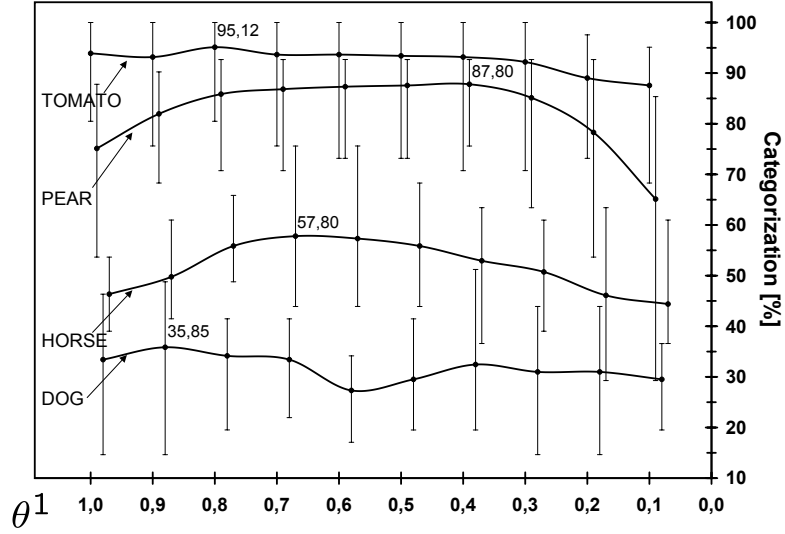
As an intermediate result the system also produces model graphs, which are the closest possible representations of a presented object in terms of memorized features. A variety of further processing can build on these graphs. The simple graph matching employed here can be replaced by the more sophisticated methods from [9, 40, 27], which should lead to increased robustness under shape and pose variations.

In the present state, the method can also be used for initialization of sophisticated but slow techniques. For instance, it can produce a coarse pose estimation followed by refinement through correspondence-field evaluation. Another promising extension will be to use diagnostics from the classification process for novelty detection and subsequent autonomous learning.

Much work remains to be done on the categorization capabilities. In our experiment we have seen that the categories employed by human cognition were not helpful to improve the categorization capability when employed to structure the recognition process. It is, however, compatible with experimental results which find that in human perception recognition of a single object instance precedes categorization [18].



(a)



(b)

**Figure 21: Categorization of Objects Using Single-Element Categories** — The averaged categorization rates within the leave-one-object-out cross-validation are displayed. Each data point was averaged over 410 single measurements. Optimal categorization performance was achieved for a well-balanced combination of the feature- and correspondence-based parts. In most cases categorization performance was clearly better than in the hierarchical case.

Another reason for the relatively poor performance in categorization experiments is that the data is much too sparse to really cover the intra-category variations. Categorization can always be improved by using additional cues like color and global shape. This would, however, also require larger databases, because much more feature combinations would need to be tested. Nevertheless, the method presented here is well suited to accommodate hierarchical categories. Their impact on categorization quality as well as methods to learn the proper organization of categories from image data are subject to future studies.

## References

- [1] M. Arentz. Integration einer merkmalsbasierten und einer korrespondenzbasierten Methode zur Klassifikation von Audiodaten. Master's thesis, Institute of Neuro Computer Science, Ruhr-University of Bochum, Germany, 2006. In preparation.
- [2] I. Biederman. Recognition-by-Components: A Theory of Human Image Understanding. *Psychological Review*, 94:115–147, 1987.
- [3] E. Bienenstock and S. Geman. Compositionality in Neural Systems. In M.A. Arbib, editor, *The Handbook of Brain Theory and Neural Networks*, pages 223–226. MIT Press, Cambridge, Massachusetts, London, England, 1995.
- [4] H. Bunke. Graph grammars as a generative tool in image understanding. In M. Nagl H. Ehrig and G. Rozenberg, editors, *Graph Grammars and their Application to Computer Science*, volume 153 of *LNCS*, pages 8–19. Springer, 1983.
- [5] M.A. Eshera and K.S. Fu. An image understanding system using attributed symbolic representation and inexact graph-matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(5):604–618, September 1986.
- [6] G. Fritz, C. Seifert, L. Paletta, and H. Bischof. Entropy based Saliency Maps for Object Recognition. In *Early Cognitive Vision Workshop*, June 2004.
- [7] B. Fritzke. A Self-Organizing Network That Can Follow Non-Stationary Distributions. In *International Conference on Artificial Neural Networks (ICANN 1997)*, pages 613–618. Springer, 1997.
- [8] R. Gray. Vector Quantization. *IEEE Signal Processing Magazine*, 1(2):4–29, April 1984.

- [9] M. Lades, J.C. Vorbrüggen, J. Buhmann, J. Lange, C. von der Malsburg, R.P. Würtz, and W. Konen. Distortion Invariant Object Recognition in the Dynamic Link Architecture. *IEEE Transactions on Computers*, 42(3):300–310, 1993.
- [10] L. Lam and S.Y. Suen. Application of Majority Voting to Pattern Recognition: An Analysis of its Behavior and Performance. *IEEE Transactions on Systems, Man, and Cybernetics — Part A: Systems and Humans*, 27(5):553–568, 1997.
- [11] B. Leibe and B. Schiele. Analyzing Appearance and Contour Based Methods for Object Categorization. In *Conference on Computer Vision and Pattern Recognition (CVPR'03)*, volume 2, pages 409–415, Madison, Wisconsin, USA, June 2003. IEEE Press.
- [12] R. Linsker. Self-Organization in a Perceptual Network. *IEEE Computer*, pages 105–117, 1988.
- [13] H.S. Loos. *User-Assisted Learning of Visual Object Recognition*. PhD thesis, University of Bielefeld, Germany, November 2002.
- [14] W.S. McCulloch and W.H. Pitts. A Logical Calculus of the Ideas Immanent in Nervous Activity. *Bulletin of Mathematical Biophysics*, 5:115–133, 1943.
- [15] B.W. Mel. SEEMORE: Combining Color, Shape, and Texture Histogramming in a Neurally Inspired Approach to Visual Object Recognition. *Neural Computation*, 9:777–804, 1997.
- [16] B.T. Messmer and H. Bunke. A new algorithm for error-tolerant subgraph isomorphism detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(5):493–504, 1998.
- [17] S.A. Nene, S.K. Nayar, and H. Murase. Columbia Object Image Library (COIL-100). Technical Report CUCS-006-96, Columbia University, 1996.
- [18] T.J. Palmeri and I. Gauthier. Visual Object Understanding. *Nature Reviews Neuroscience*, 5:291–304, April 2004.
- [19] D.I. Perret, P.A.J. Smith, D.D. Potter, A.J. Mistlin, A.S. Head, and A.D. Milner. Visual Cells in the Temporal Cortex Sensitive to Face View and Gaze Direction. *Proceedings of the Royal Society B*, 223:293–317, 1985.
- [20] M. Pöttsch, T. Maurer, L. Wiskott, and C. von der Malsburg. Reconstruction from Graphs Labeled with Responses of Gabor Filters.

- In C. von der Malsburg, W. von Seelen, J. Vorbrüggen, and B. Sendhoff, editors, *Proceedings of the ICANN 1996*, pages 845–850, Berlin, Heidelberg, New York: Springer, 1996.
- [21] M. Riesenhuber and T. Poggio. Models of Object Recognition. *Nature Neuroscience*, 3:1199–1204, November 2000.
- [22] F. Rosenblatt. The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain. *Psychological Review*, 65:386–408, 1958.
- [23] P.A. Schmidt and G. Westphal. Object Manipulation by Integration of Visual and Tactile Representations. In B. Mertsching, M. Lappe, H. Neumann, C. von der Malsburg, J. Müsseler, and G. Sagerer, editors, *Proceedings of the Dynamic Perception Workshop (DP 2004)*, Tübingen, Germany, November 2004.
- [24] L.B. Shams. *Development of Visual Shape Primitives*. PhD thesis, University of Southern California, 1999.
- [25] C.E. Shannon. A Mathematical Theory of Communication. *Bell Systems Technical Journal*, 27:623–656, 1948.
- [26] L.G. Shapiro and R.M. Haralick. Structural descriptions and inexact matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 3(5):504–519, 1981.
- [27] Andreas Tewes. *A Flexible Object Model for Encoding and Matching Human Faces*. PhD thesis, Physics Dept., Univ. of Bochum, Germany, January 2006.
- [28] S. Thorpe, D. Fize, and C. Marlot. Speed of Processing in the Human Visual System. *Nature*, 381:520–522, 1996.
- [29] S. Thorpe and M.F. Thorpe. Seeking Categories in the Brain. *Neuroscience*, 291:260–263, 2001.
- [30] I. Ulusoy and C.M. Bishop. Generative Versus Discriminative Methods for Object Recognition. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2005)*, San Diego, California, USA, volume 2, pages 258–265. IEEE Press, 20–26 June 2005.
- [31] M. Vidal-Naquet and S. Ullman. Object Recognition with Informative Features and Linear Classification. In *Conference on Computer Vision and Pattern Recognition (CVPR’03)*, pages 281–288, Madison, Wisconsin, USA, June 2003. IEEE Press.

- [32] C. von der Malsburg. The Correlation Theory of Brain Function. Internal report 81-2, Max-Planck-Institute for Biophysical Chemistry, Department of Neurobiology, 1981.
- [33] C. von der Malsburg. Pattern Recognition by Labeled Graph Matching. *Neural Networks*, 1:141–148, 1988.
- [34] C. von der Malsburg. The Dynamic Link Architecture. In M.A. Arbib, editor, *The Handbook of Brain Theory and Neural Networks*. MIT Press, Cambridge, Massachusetts, London, England, second edition, 2002.
- [35] C. von der Malsburg and K. Reiser. Pose Invariant Object Recognition in a Neural System. In F. Fogelmann-Soulié, J. C. Rault, P. Gallinari, and G. Dreyfus, editors, *International Conference on Artificial Neural Networks (ICANN 1995)*, pages 127–132. EC2 & Cie, Paris, France, 1995.
- [36] G. Westphal. Classification of Molecules into Classes of Toxicity. Technical report, Dr. Holthausen GmbH, Bocholt, Germany, 2004.
- [37] G. Westphal. *Feature-driven Emergence of View-tuned Object Representations*. PhD thesis, University of Lübeck, Germany, 2006. In preparation.
- [38] G. Westphal and R.P. Würtz. Fast Object and Pose Recognition Through Minimum Entropy Coding. In Josef Kittler, Maria Petrou, and Mark Nixon, editors, *17th International Conference on Pattern Recognition (ICPR 2004)*, volume 3, pages 53–56, Cambridge, UK, August 2004. IEEE Press.
- [39] L. Wiskott. *Labeled Graphs and Dynamic Link Matching for Face Recognition and Scene Analysis*. PhD thesis, Faculty of Physics and Astronomy, Ruhr-University of Bochum, D-44780 Bochum, Germany, 1995.
- [40] L. Wiskott, J.-M. Fellous, N. Krüger, and C. von der Malsburg. Face Recognition by Elastic Bunch Graph Matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):775–779, 1997.
- [41] I.H. Witten and E. Frank. *Data Mining: Practical Machine Learning Tools and Techniques with JAVA Implementations*. Morgan Kaufmann, 2000.
- [42] R.P. Würtz. Object Recognition Robust Under Translations, Deformations, and Changes in Background. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):769–775, 1997.